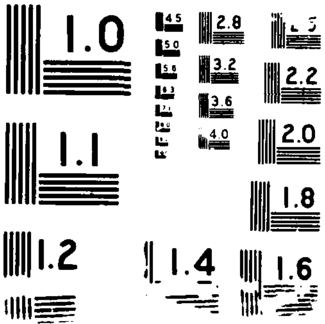
AD-A194 024 THE P-3 SCHEDULING SUPPORT SYSTEM (P-3 S3)(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W B ANDERSON MAR 88 1/2

UNCLASSIFIED F/G 1/3.4 NL


1.0
4.5
5.0
5.6
2.8
3.2
3.6
4.0
2.2
2.0
1.1
1.8
1.2
1.4
1.6

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE P-3 SCHEDULING SUPPORT SYSTEM (P-3 S^3)

by

William B. Anderson

March 1988

Thesis Advisor: Barry A. Frew

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

[illegible]

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
Distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Postgraduate School

6b. OFFICE SYMBOL *(If applicable)*
Code 54

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS *(City, State, and ZIP Code)*
Monterey, California 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, California 93943-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
THE P-3 SCHEDULING SUPPORT SYSTEM (P-3 S^3)

12. PERSONAL AUTHOR(S)
Anderson, William B.

13a. TYPE OF REPORT
Master's Thesis

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
1988 March

15. PAGE COUNT
148

16. SUPPLEMENTARY NOTATION
"The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government."

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Management Information System; Software Development; Flight Schedule

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

A P-3 Scheduling Support System (P-3 S^3) is a Management Information System (MIS) that was designed using structured techniques. Structured analysis was used to determine the functionality and data requirements. Computer Assisted Systems Engineering (CASE) tools were used to document the analysis and design. The system was designed to be implemented in dBase III Plus, a data base management tool developed by Ashton Tate. P-3 S is designed to run on a micro-computer with the MS-DOS operating system. It provides real-time access to historical data and provides suggested personnel assignments to the user. The design provides for faster flight schedule generation and prevention of conflicting schedule events.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Prof. Barry A. Frew

22b. TELEPHONE *(Include Area Code)*
(408) 646-2924

22c. OFFICE SYMBOL
Code 54Fw

83 APR edition may be used until exhausted.
All other editions are obsolete

U.S. Government Printing Office: 1986—606-24.

The P-3 Scheduling Support System (P-3 S^3)

by

William B. Anderson
Lieutenant Commander, United States Navy
B.S, Iowa State University, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1988

Author: William B Anderson
William B. Anderson

Approved by: ______________________
Barry A. Frew, Thesis Advisor

____
John B. Isett, Second Reader

____
David R. Whipple, Jr., Chairman
Department of Administrative Sciences

____
James M. Fremgen, Acting
Dean of Information and Policy Sciences

ABSTRACT

A P-3 Scheduling Support System (P-3 $S^3_{\sqcup}$) is a Management Information System (MIS) that was designed using structured techniques. Structured analysis was used to determine the functionality and data requirements. Computer Assisted Systems Engineering (CASE) tools were used to document the analysis and design. The system was designed to be implemented in dBase III Plus, a data base management tool developed by Ashton Tate. P-3 $S^3_{\sqcup}$ is designed to run on a micro-computer with the MS-DOS operating system. It provides real-time access to historical data and provides suggested personnel assignments to the user. The design provides for faster flight schedule generation and prevention of conflicting schedule events.

(Keywords: [illegible] [illegible], patrol [illegible] [illegible].

TABLE OF CONTENTS

I. INTRODUCTION

A. SQUADRON ORGANIZATION

A typical Patrol Squadron has six departments: Administration, Operations, Maintenance, Tactics, Training and Safety. One of the primary tools with which the squadron training and operations requirements are managed is the Flight Schedule. In developing the Flight Schedule the departments have the following responsibilities:

* Each workday the Operations Department determines how many flights should be flown the following day and assigns the crews to those flights.

* The Training Department tracks the individual crewmember's training history to determine what training he should receive to qualify at his assigned crew position.

* The Training Department and the Operations Department work together to optimize the training accomplished on each flight scheduled.

* The Safety Department schedules flight evaluations and ground training for the crewmembers through the Operations Department.

* The Maintenance Department schedules ground training for Inflight Technicians and Ordnancemen.

* The Administration Department sometimes schedules Administration events such as All Officer Meetings on the Flight Schedule.

High operational readiness is the squadron's main objective. Reduced resource availability requires that the squadron conduct its training as efficiently as practicable. Generation of the daily squadron flight schedule requires significant information transfer between the departments. This information is currently stored in numerous (often redundant) files throughout the departments. Gathering, collating, prioritizing and evaluating this information is manpower intensive. Quick revision of the flight schedule using the current manual system is difficult. A common complaint with the manual system occurs when personnel are simultaneously scheduled for conflicting events.

The typical Navy P-3 squadron has up to twelve flight crews of five officers and seven or eight enlisted personnel each. The process of creating a daily flight schedule for an operational P-3 squadron requires between fifty and two hundred manhours per week depending on operational tempo. Operational tempo changes rapidly and often for a Patrol Squadron.

While not deployed, each squadron takes its turn standing the ready alert cycle. During the ready alert cycle, the squadron must routinely fly numerous short notice "real world" operational flights in addition to its normal training flights. When not in an operational cycle, the operational tempo is generally more relaxed. While deployed, a squadron must remain ready to fly short notice operational flights. Therefore the flight schedule must account for several potentially conflicting factors.

In accordance with the navy P-3 Personal Qualification Standards, each unqualified aircrewman (officer and enlisted), must fly predetermined positional qualification training flights. Each officer must have yearly flight physicals within thirty days of his birthday, and take written and flight tests in NATOPS[1] and instrument flight. PATROL WINGS PACIFIC/ATLANTIC INSTRUCTIONS require each crew to maintain proficiency in all missions of the P-3 aircraft. These areas are evaluated during qualification flights. To be qualified on one of these qualification flights, certain members of the crew must be aboard the aircraft in their assigned aircrew positions. Sickness, emergency leave, crew rest and unplanned operational flights make the manual process of determining flight schedules difficult.

A Management Information System for flight schedule generation would provide much needed administrative and managerial support for both the Operations Department and the Training Department. Each west coast Patrol Squadron is in the process of receiving two Zenith Z-248 microcomputers and Database III Plus software from Commander, Naval Air Force Pacific Fleet. The two

[1] NATOPS (Naval Air Training and Operating Procedures Standardization)

Zenith micro-computers will be shared by the Maintenance, Training, Operations and Administration Departments. The standard database software package for all aviation squadrons in the Pacific is expected to be dBase III Plus.

This thesis contains a completed Functional Description and Design Specification for a management information system to assist in the generation of P-3 squadron flight schedules. It consists of a system designed to be implemented on the squadron Zenith Z-248 microcomputer using the dBase III Plus database management language.

B. GOALS AND OBJECTIVES

The major objective of this thesis was to evaluate the structured methodologies, during analysis and design of the P-3 Scheduling Support System. An initial constraint was that it would be implemented in dBase III Plus. The appendices of this thesis provide the Functional and Design Specifications from which a complete MIS can be implemented and maintained.

The P-3 squadron is one of the most complicated naval aviation squadrons to schedule because of the multitude of personnel flying in the different crew positions. The logical and physical designs for other aviation community Flight Schedule Generators should involve substitution of appropriate modules to this design.

A functional Flight Schedule Generator was designed to provide the Operations and Training Departments an environment in which current manual operations, duplications and training record maintenance can be significantly reduced. It will simplify the determination of who is available to be scheduled and ensure that no one is simultaneously scheduled for mutually exclusive events. The outputs include a Flight Schedule, operational event schedule flows, reports for keeping track of flight hour allocations and user designed reports from database queries. The following tasks will be performed by the Flight Schedule Support System MIS:

Daily Processing
* Crewmember Entry Modification and Deletion
* Crewmember Personal Data modification
* Crewmember Training History Modification
* Crewmember Training Syllabus Modification
* Crewmember Availability Data Modification
* Flight Schedule Generation

General Administration
* Prepare Crew Listing
* Temporal Operational Commitment Processing
* Long Range Training Plan Processing
* Prepare Flight Hour Status Graph
* Prepare Operation Event Schedule Flow.

The predominant improvement expected to be provided by the P-3 Scheduling Support System is the speed with which it will permit flight schedules to be generated. Additionally, it will ensure that crewmembers are not scheduled for two events simultaneously. It will allow a significant reduction of redundant data being maintained by Training and Operations Department personnel. Complex ad hoc queries can be made and the answers provided in seconds. Elimination of redundant data maintained by numerous personnel in the Training and Operations Department, will improve the integrity and correctness of the data.

II. THEORY AND METHODOLOGY

The classic software system life cycle includes the following:

- Problem definition
- Feasibility Study
- Analysis
- System Design
- Detailed Design
- Implementation
- Maintenance [Ref. 1:p. 17]

This was the methodology used to develop the P-3 Scheduling Support System. This chapter describes the system design theory and methodology used to analyze and design the P-3 Scheduling Support System.

A. BACKGROUND

Boehm discusses the increasing cost of acquiring and maintaining software since the computer has become a commonly used business tool. While "Off the Shelf" microcomputer software packages can commonly be purchased inexpensively, custom software developed for all types of computers has become a high budget item. Software lifecycle maintenance costs are higher than the cost of the original development. Reduction of this maintenance cost should therefore receive significant management attention. The structured methodologies improve communication between the user, the analyst and the programmer. [Ref. 2:p. 4] The first step in the software development process is problem definition.

B. PROBLEM DEFINITION

Having been involved with P-3 flight schedules, from both the squadron and wing perspectives, the author observed the tedious manual effort that was required to generate one (or more) flight schedule(s) each day. This process involves maintaining, sorting and evaluating large volumes of data. The highest level of squadron flight schedule automation observed at NAS Moffett Field was one squadron using LOTUS 123 for generating operational event flows. Other "automated" squadrons use a microcomputer word processing package to convert a hand written rough into a typed smooth flight schedule. In both cases, record keeping is done manually, often in redundant files.

The scheduling "cut and paste" process is generally done on a plexiglass grease board. The process is labor and time intensive and intuitively lends itself to automation. The problem then is that scheduling flight crews is a time and labor intensive, complex process which requires evaluation of a large amount of data. There is a need to make this process quicker and easier. Determining whether automation is feasible is the next step.

C. FEASIBILITY STUDY

Davis discusses three types of feasibility: technical, financial, and political:

* Technical feasibility answers the question, can it be done.

* Financial feasibility answers the questions: can it be delivered at or below the price we can afford to pay, and will the solution be worth its cost?

* Political feasibility asks the question, "can it be done here"? [Ref. 1:p. 39]

There are currently several commercial scheduling software packages available. Unless the P-3 flight scheduling process was several orders of magnitude more complex than other scheduling problems, it should be technically feasible. The financial feasibility question must be answered by Commander, Naval Air Force U.S. Pacific Fleet. Further development of the system should not be undertaken unless a cost benefit analysis shows that the system is worth the cost of implementation and maintenance. A cost benefit analysis was not conducted as part of this thesis. This thesis evaluates the structured methodologies in the development of Functional and Design Specifications for programming the system in dBase III. From the squadron perspective, the answer to the political feasibility question has been an unqualified "yes!" Each operations officer interviewed has been very enthusiastic about automating this process. This project is certainly politically feasible.

Assuming that the implementation will be financially feasible, the next steps in the classic software lifecycle are system analysis and design.

D. SYSTEM ANALYSIS AND DESIGN

Structured analysis and design commonly use a set of communication tools including: Data Flow Diagrams, Data Dictionaries, Mini-specifications, Structure Charts, Data Dictionaries/Directories and Module Specifications. Several of these tools were used to define the Functional and Design Specifications for this thesis. Page-Jones discusses the Yourdon methodology of structured design in terms of these communication tools:

* The Data Flow Diagram (DFD) can be used to validate the logical decomposition of the function that the program is to automate. The DFD also communicates the logical functions to the programmer.

* A Data Dictionary is a standardization tool for data, a description of the data structure formats and a description of data use.

* Mini-specifications are process descriptions of the lowest level modules of the DFD. They describe the logical functionality each module of the DFD represents. This is usually done either in structured english or pseudocode, both of which look similar to a programming language.

* Structured design of software uses the Structure Chart, Data Dictionary/Directory and Module Specifications to describe the physical design of

the software. The Structure Chart is used as a graphical communication tool between the Analyst and the Programmer. It describes the physical design for the program. The data that is communicated between procedures or subroutines of the program is graphically displayed on the Structure Chart.

* The Data Dictionary/Directory is similar in structure and function to the logical Data Dictionary, except that it emphasizes physical storage and indicates where each data item is used throughout the program.

* The module specifications are analogous to the mini-specifications except that they show the programmer how the module they describe should perform its function. [Ref. 2:pp. 337-350]

Structured design allows development of programs that simultaneously have high cohesion and low data coupling. Davis and Olson discuss cohesion as a measure of the number of different functions a procedure or subroutine incorporates. High cohesion implies that the program contains separate procedures for each functional area. Data coupling is a measure of the commonality of data used by different procedures or subroutines. Development of programs which have high cohesion and low data coupling permit easier maintenance. If functionality needs to be changed, modify only the module that provides that functionality. [Ref. 3:pp. 279-283]

One problem with the Yourdon design methodology is that it does not deal with Database Management Systems (DBMS) environments. Yourdon did not address data redundancy and data integrity during insertion, modification and deletion. [Ref. 4:p. 286]

E. DATABASE CONSIDERATIONS

Why use a DBMS for the P-3 Integrated Flight Schedule System? According to Kroenke, a DBMS provides multiple views or different looks at the stored data. This means that more usable information can be produced from a given amount of data. The DBMS stores a description of data formats, stores the data and retrieves the data. This provides data independence for the programs using this data. The data therefore does not need to be redefined in each module or program. Prior to database programming, each application had its own data files. [Ref. 4:p. 4] This is similar to manual systems currently used by the squadrons for flight documentation and scheduling. The Training Department maintains numerous files containing information about training that has been completed. The Operations Department maintains files which contain some of the same data as the Training Department files. If the files of one department are changed and the files of the other are not, then data integrity is lost. A DBMS will allow both departments to enter, modify or delete shared data so that data integrity is maintained.

A disadvantage of using a computerized DBMS system is that it makes the data more vulnerable to system failure [Ref. 4:p. 415]. If the computer system fails, the scheduling process can be more difficult than it is under the manual system. The data required to make scheduling decisions is only available in the computer. For this reason, the data should be routinely backed up, so that the scheduling system could be quickly rebuilt on another computer if necessary.

A DBMS characteristically provides two programming capabilities. A Data Definition Language (DDL) and a Data Manipulation Language (DML). The DDL is used to define the data structures (eg., Character, Integer, Real, Logical) and the DML is the programming language for applications interfacing with the Data Base(s). The logical design of a DBMS approach should include the normalization of anticipated data files. When un-normalized files have data inserted, modified or deleted problems can occur. Normalization involves storing the data in files which conform to the following normal forms:

* First Normal Form--all fields are single valued, repeating groups are not allowed.

* Second Normal Form--key fields are the fields in a record in which the data (which is unique to that record) uniquely identifies that record. A relation is in second normal form if all the non-key fields in that record are uniquely identified by the key field(s)

* Third Normal Form--the key field(s) uniquely identify all non-key fields without having to be related to another non-key field (transitivity) [Ref. 4:pp. 286-306]

Kroenke described four additional normal forms which are more theoretical than practically applicable. They are: Boyce-Codd Normal Form, Fourth Normal Form, Fifth Normal Form and Domain Key Normal Form. These additional normal forms will not be discussed further.

Data hiding (data known to one module is unknown in others unless that data has been passed to it as a parameter) is not possible with dBase III Plus. A variable instantiated in one dBase III program or procedure is globally available unless it has been cleared from memory. The structured analysis and design techniques advocated by Yourdon are generally valid for analyzing and designing programs which will be implemented in dBase III Plus. The Yourdon structured design methodology does not address the capabilities of the dBase III Plus environment. With dBase III Plus the user can generate queries of the data files, he can edit, append, modify structure and delete data in those files. This can be done either through an application written in the DML or directly from the dBase III environment. Prior to designing applications that will comprise the P-3 Scheduling Support System, the manual system was analyzed to determine what logical functionality existed.

F. APPLICATION OF THE STRUCTURED METHODOLOGY

The Yourdon methodology was first used to analyze the current manual scheduling system used by several of the NAS Moffett Field, P-3 squadrons. A data driven approach was selected for analysis of the manual squadron flight scheduling systems. Data driven analysis was chosen because although each squadron conducted the scheduling process slightly differently, the output Flight Schedules and other associated reports were similar [Ref. 1:p. 135]. All data exiting the manual flight scheduling system, was necessarily either entered into the system or generated by the system. By evaluating the individual data items of the outputs, it was possible to derive the functions required to generate those data items. A function driven approach was not chosen because the manual flight scheduling process was not readily decomposable into functional areas and the use of a DBMS required focusing on data structures and data flows.

The Yourdon methodology for structured design had to be modified for use with a DBMS. A typical structure chart shows data passage between superior and subordinate modules of a program. Series of menu programs and data manipulation programs often comprise the structure of dBase III programs. The menu programs control which data manipulation program is called.

Each data manipulation program separately addresses required data files. During system design a partial prototype was built and shown to Patrol Squadron Forty Operations Department personnel. This was done to test the validity of the initial analysis. Although they were quite pleased with the prototype, and offered sound suggestions on improvement, development of a prototype early in the lifecycle caused serious side effects. The initial prototype determined which pilots should have the highest priority for training flights. Additionally, it contained pilot training historical files and a personnel file. Once the prototype was developed, it was extremely difficult to design a system which was not influenced by the logic and structure of the prototype. The initial system design began as an extention of the prototype. Extention of the prototype resulted in an inefficient design for the entire system. A poorly designed prototype (which it was) caused a lot of wasted development and design time by infecting the logical definitions and design of the system.

G. PROTOTYPING

A software project which involves prototyping must be managed. Historical problems with software prototyping include:

- Inadequate system analysis--errors in analysis that are not found until testing or post delivery are several magnitudes more costly to correct than if found during analysis.

- User management wants to hold cost down, so decide to use the prototype rather than wait for the operational version

- Gold Plating or overkill--if management can not control the project, it is difficult to determine when the project is finished. If the scheduled completion date was too generous, the prototype may receive additional bells and whistles that do not effectively improve the software.

- Senior managers are inappropriate team members for prototyping, because they do not have the time to devote to effective participation.

- Lack of project discipline causes confusion and wasted effort because time and effort is not effectively managed.

- Documentation is left to last. Historically software documentation is generally poor. One of the goals of prototyping is more rapid development this should not be at the expense of system documentation. Good documentation and good design will provide significant future maintenance cost savings. [Ref. 5:p. 2]

The old cliche "If it ain't broke don't fix it" may not apply when considering prototyping. The most serious problem with a prototyping methodology is the management of the prototype development. Pressman suggests that it is more difficult to predict cost and schedule for prototyping projects. This is because it is more difficult to put management boundaries on a prototyping project. Management should realize that to improve the maintainability of the software, the prototype needs to be decomposed, analyzed, redesigned and reprogrammed. To many in management this sounds like the prototype is wasted, when in fact direct use of the prototype might be much more costly because:

- the design or programming is inefficient
- the prototype is difficult to test
- difficult to correct errors, enhance or transport the prototype [Ref. 6:pp. 148-160]

Cesena and Jones describe how the Army has begun to manage their prototyping efforts using a "Contemporary Life-Cycle Development Methodology". In this methodology, prototype management is emphasized. Quality assurance reviews play a large part of this management. Like the conventional software lifecycle, the Contemporary Life-Cycle Development methodology begins with concept exploration, a proposal and a feasibility study. The Requirements Definition stage includes conceptual planning and a broad high level functional specification. This high level functional specification describes module interfaces and system boundaries. During this phase the developer and the user form a team to determine the system requirements.

As each part of the system is developed, it is reviewed by the developer-user team. The design phase includes logical design of the data base (if required) and application design. The programming phase includes the physical design and implementation of the data base and applications. At the end of the programming phase, the system is demonstrated for the user. At this time the final definition of user requirements is documented.

The system enters testing and validation. It is delivered to the user for a site test. During deployment, operation and maintenance new requirements are generated and adapted in a configuration control atmosphere. [Ref. 5:p. 6]

Cesena and Jones describe their managed prototyping methodology as:

> . . . a methodology which recognizes differences in project related factors, in contrast to the traditional approach which forced all projects to use a single development strategy. For highly structured systems with requirements that can be determined in a straight forward process, a linear strategy with minimal iteration can be used. Where uncertainty is relatively high (large multiple-user systems or applications new to users or the developer), a life cycle structure with embedded prototyping is an available option.
>
> Systems or applications with high levels of requirements uncertainty can apply iterative approaches such as prototyping and systems simulation. Examples of high uncertainty are executive decision support, command and control and other unstructured applications for which it is difficult to specify requirements in advance (or requirements are expected to change significantly during development). A contemporary Life-Cycle Development Methodology is, in other words, a contingency model that permits the selection of a development strategy consistent with the level of uncertainty about user requirements or technology. [Ref. 5:p. 7]

Effective management of prototyping adapts techniques from the classical software lifecycle management processes. Prototyping is often used to develop complex systems such as Decision Support Systems and Expert Systems.

H. DSS OVERVIEW

A Decision Support System is a computer based tool to assist managers in making semi-structured decisions. The computer can conduct the numerical or logical analysis but the manager uses his expertise, judgment and insight to make the decision. A DSS is more beneficial to the manager when the decisions are highly repetitive [Ref. 3:p. 371]. An optimization DSS would be most beneficial to the P-3 Interactive Flight Scheduling System. This type of DSS provides recommendations to the decision maker to maximize or minimize a specific objective [Ref. 7:pp. 89-109].

Rowe describes an Expert Support System as a DSS into which the experts specialized knowledge base is programmed. A good Expert Support System will explain why it recommends certain actions. Expert Support Systems use a knowledge data base and decision rules. The knowledge data base is a description of the problem in a logically coherent and formal manner. This description includes known facts and knowledge about the problem as well as specified goals. The decision rules give the Expert Support System the capability to infer facts and conclusions from other facts. Two commonly used languages for programming Expert Support Systems are LISP and Prolog [Ref. 8:p. 8]. LISP and Prolog are object oriented languages. Procedural languages like Pascal, Fortran, Cobol, C, and dBase III define an algorithm (procedure) to solve the problem. LISP and Prolog do not use procedures. They use data bases of objects and rules describing relationships between the objects. [Ref. 9:pp. 4-9] Decision Support Systems and Expert Systems which use object oriented languages generally are developed through prototyping. One of the problems of managing prototype development is the sparseness of documentation.

Ensuring that the logical and physical designs are fully documented has always been a problem for management. This is especially true if the designs are often changing due to differing requirements or due to the evolution of a prototype through evolving designs. The Computer Aided Software Engineering (CASE) tools permit easier documentation of the software development process.

I. USING COMPUTER AIDED SOFTWARE ENGINEERING TOOLS

The majority of the analysis and design of the P-3 Schedule Support System was graphically documented using a Computer Assisted Software Engineering (CASE) tool called DESIGNAID. CASE tools such as DESIGNAID and EXCELERATOR are powerful effort and time saving software products. Both allow the use of a mouse for pointing, screen manipulation and drawing the graphical Data Flow Diagrams and Structure Charts. While the CASE tool user creates a DFD or Structure Chart, he can simultaneously create the Data Dictionary. This includes adding data definitions, occurrence and structure information.

Both DESIGNAID and EXCELLERATOR allow the user to create report forms which can to be used to verify output requirements of the system under design. This is similar to the prototyping process used to validate Functional Specifications. After the DFD or Structure Chart is drawn, the CASE tool can automatically validate it. Validation of a diagram includes making sure that all the objects in the diagram are defined using the proper syntax. Validation also ensures that the DFD is drawn to "Yourdon standards." Once the diagram is validated, all occurrences of the data flows, external entities, processes, data stores and functions are documented in the Data Dictionary. An important capability of the CASE tool is the ability to balance DFDs.

> Because of the complexity of most systems, every component cannot be included in one single diagram. Instead, the system is gradually broken down into several diagrams that represent increasing levels of detail.
>
> When you balance your data flow diagrams you are checking the data flowing into and out of diagrams to ensure that the net inputs and outputs on each level are equal. Because some objects have structures, one input or output may represent several other objects.
>
> The balancing process checks to make sure that the structures which make up objects are all accounted for.[2]

The real benefit of using a CASE tool is the ease and speed with which a logical or physical design can be generated and modified or scrapped and redone. Without a CASE tool, this is a slow and tedious process.

The major drawback of both CASE tools was the inability to customize their data dictionaries. The data dictionary, mini-specifications and module specifications for this system were generated with a CASE tool and then modified with a word processor. During each phase of system development, management needs controls and milestones to evaluate progress and system quality. Test planning and test management are an important part of software development, but are not within the scope of this thesis. Quality assurance

[2] Nastec Corporation CASE 2000 Design Aid User Guide

procedures should be incorporated into the controls of each step of the software development process.

J. QUALITY ASSURANCE DURING SYSTEM DESIGN

Pressman defines Software Quality Assurance as:

> Software quality assurance (SQA) is an "umbrella activity" that is applied throughout the software engineering process. SQA encompasses: (1) analysis, design, coding, and testing methods and tools; (2) formal technical reviews that are applied during each software engineering step; (3) a multitiered testing strategy; (4) control of software documentation and the changes made to it; (5) a procedure to assure compliance with software development standards and (6) measurement and reporting mechanisms. [Ref. 6:pp. 433-436]

Pressman also states that a large majority of the errors introduced into software are introduced during the design phase. The Quality Assurance tools that most effectively discovers this type of error is a formal review technique, the walkthrough [Ref.6:p.440].

Yourdon states that the formal technical reviews are designed to find flaws in the logic, design or implementation of the software. They are used to ensure that the software design meets the functional requirements. Formal technical reviews help ensure the software is developed to applicable standards. Formal technical reviews help train less experienced personnel in the analysis, design and implementation of software from different perspectives. [Ref. 10:pp. 171-186]

Yourdon describes four types of walkthroughs:

* Specification Walkthroughs -- to look for problems, inaccuracies, ambiguities, and omissions in the system specification.

* Design Walkthroughs -- to look for flaws, weaknesses, errors, and omissions in the architecture of the design before code is written.

* Code Walkthroughs -- review of the code written by each programmer by other programmers.

* Test Walkthroughs -- to ensure the adequacy of the test data for the system. [Ref. 10:p. 174]

One question that the software developer must answer is how large a QA effort is economical:

> Proof techniques should be used in situations where the operational cost of a software fault is very large, that is, loss of life, compromised national security, major financial losses. But if the operational cost of a software fault is small, the added information on fault-freedom provided by the proof isn't worth the investment. [Ref. 11:p. 298]

Demarco defines software quality as the sum of the defect diagnosis and correction costs divided by the program volume. He states that the average American produced code has 10 to 50 defects per thousand lines of executable code, while the Japanese produce code with an error rate of 0.2 defects per thousand lines of executable code. This disparity reflects the acceptance of software defects as a "fact of life" by Americans and the unacceptability of software defects to the Japanese. A large part of the Japanese success may be the result of superior analysis and planning. [Ref. 12:pp. 205-211]

The logical and physical design of the P-3 Scheduling Support System underwent a series of technical reviews. The initial DFDs were submitted for review. As a result of the review process, the entire logical design of the system changed during the evolution to the final DFDs.

Following development of an acceptable logical model, the mini-specifications and data dictionary were developed. These documents underwent technical review and were subsequently enhanced. The structure charts were developed almost directly from the DFDs, mini-specifications and data dictionary. The structure charts too underwent technical review. After implementation and delivery to the fleet, the software will undergo the final phase of the software lifecycle, maintenance. The structured analysis and design methodology produces software which is easier to maintain than software that is generated otherwise [Ref. 6:p. 529].

K. MAINTENANCE OF THE SOFTWARE

The term maintenance is commonly used to include error correction, incorporation of additional capabilities and transformation of the software for use with different hardware. Psychologically, the job of a maintenance programmer is difficult, and not just because the code was developed by someone else. Management wants all defects fixed "yesterday". Software errors can make the system unusable, require lengthy reprocessing of previously completed applications and loss of customer good will. [Ref. 6:pp. 525-551]

Structured analysis, design and programming should be followed by structured maintenance:

> If the only available element of a software configuration is source code, maintenance activity begins with a painstaking evaluation of the code, often complicated by poor internal documentation. Subtle characteristics such as program structure, global data structures, system interfaces, performance, and/or design constraints are difficult to ascertain and frequently misinterpreted. The ramifications of changes that are ultimately made to the code are difficult to assess. Regression tests (repeating past tests to assure that modifications have not introduced faults in previously operational software) are impossible to conduct because no record of testing exists. We are conducting unstructured maintenance and paying the price (in wasted effort and human frustration) that accompanies software that has not been developed using a well defined methodology. [Ref. 6:p. 551]

Maintenance of the software is the main reason why structured analysis and design is so important. The cost of maintenance has risen to the point that maintaining software is more costly than developing it. Does anyone include the future expected maintenance cost in the original cost benefit analysis? Probably very few. Variables such as how long it would be used before replacement and how many enhancements would be incorporated into the baseline, make such a calculation difficult. Boehm's COCOMO economic analysis product does allow that cost estimation, but again the user must enter the estimations of the size of the changes expected [Ref. 11:pp. 129-134].

L. SUMMARY

The structured methodologies for software development provide improved management of software throughout its lifecycle. Management control is improved by adding check-points where progress can be reviewed. System requirements are validated with the user through the DFD, Mini-specifications and Data Dictionary. The analyst designs the system from the DFDs, Mini-specifications and Data Dictionary. The design is documented with the Structure Chart, Module Specification and Data Directory. The programmer uses these to implement the design. With certain modifications, dBase III Plus can implement the design using structured techniques.

III. MIS SYSTEM

A. ANALYSIS INTRODUCTION

1. Initial Analysis

Analysis began with the initial visit to several Patrol Squadrons and Patrol Wing TEN at Naval Air Station, Moffett Field, California. An initial series of discussions introduced the project to the squadron Operations Department and Training Department personnel. All personnel contacted were eager to assist, indicating that an automated flight scheduling system would significantly simplify their work. Subsequent visits were scheduled, and squadron personnel were asked to gather the documentation from which the data requirements could be determined. Flight schedules, monthly training plans, weekly training plans, long range training plans and crew lists were all obtained.

OPNAV 3710/4 Naval Aircraft Flight Record "Yellow Sheets" provide flight data to the Operations Department. Yellow Sheet flight data includes date of the flight, personnel on board, total flight time, official land time and number of approaches and landings during the flight. Each squadron uses a self generated Flight/Training Recap Sheet to document flight training and ground training conducted during each Flight Schedule event. This form repeats much of the information in the OPNAV 3710/4 "Yellow Sheet". It contains additional data including readiness qualifications obtained, Personal Qualification Standardized Training (PQS) accomplished and an area to write a mission synopsis.

The Moffett squadron Flight Schedules all had the same general appearance and content. A P-3 squadron flight schedule first displays some of the Squadron Watch Bill data and then a description of what "Cycle" (Training, Admin, Duty or Operational) each crew was scheduled for that week. The flight events section contains: the flight schedule event number, the preflight time, take off time, the expected flight duration and the expected land time. If a tactical crew was not assigned, a "make-up" crew is formed. This is generally the case for pilot training flights. For "make-up" crews, each crew member is listed separately. For tactical crew events, the crew number, the Tactical Coordinator, the Plane Commander and any additions or deletions from the tactical crew are annotated. A ground training section follows the flight events section. The ground training events follow the flight events. The ground training events indicate: the starting time of the event, who is to attend, what the training is to accomplish and who will instruct. Notes at the end of the flight schedule give additional information about the events.

Patrol Squadron FORTY furnished a Weekly Training Plan, which was representative of the weekly training plans used by the other Moffett squadrons. The weekly training plan, had a graphical presentation of each crew's schedule for the week, including school and trainer events. It contained a prioritized list of proposed pilot training for the week and anticipated crew flight events for the week. A detailed daily ground training schedule followed.

The Monthly Training Plan furnished by Patrol Squadron FORTY SEVEN was representative of the Monthly Training Plans used by the other Moffett squadrons. It included the assignment of action officers to major squadron events for the following three months, the planned flight hour allocation and crew assignment status (operational, training, admin, duty, leave) by the week. A graphical calendar of anticipated flight events indicated the expected flight hour requirements for each day. Each unqualified crew member was listed with the qualification training he was expected to accomplish that month. Aircrew weapons load requirements followed a weapons training section. The remainder of the Monthly Training Plan documented tactical training, acoustic and nonacoustic operator training and schools scheduled. The last schedule in the Monthly Training Plan was a revised duty calendar for the aircrew.

The Long Range Training Plans were all classified, so discussion of the Long Range Training Plans has been generalized to allow an unclassified presentation. The Long Range Training Plan summarizes the anticipated training requirements for a squadron for the following year. These training requirements include those for flight crewmembers and ground personnel. For example all the anticipated schools that squadron personnel need to attend are listed. Each unqualified aircrew member's expected training schedule is included in this document.

2. System Design Constraints

The two most important design objectives were to develop a useful system and a usable system. The usefulness of the system will be based upon whether the generated flight schedule is quick, correct and in the correct format. The usability of the system will be measured on two levels. The menu driven system should provide an easy logical view of the system and it should teach the novice user the basics of generating a squadron flight schedule. Once the user is familiar with using the MIS, he should become familiar with the Flight Schedule generation procedures and logic.

The MIS design supports files and processes which maintain online historical records (ie., Update Pilot Training History Data). This was done to provide the user with an online query and report generation capability. Manual historical data tracking and report generation prevents the effective use of the training records for making real-time decisions.

B. OVERVIEW OF THE COMPLETED DESIGN

The Data Flow Diagrams (DFDs) contained in Appendix A represent the graphical view of the logical definition of the P-3 Scheduling Support System MIS. The DFDs divide the system into logical subsystems enabling understanding of the whole system as a sum of its parts. The DFDs provide a top down partitioning, from the complete system to the independent components. Each sub-level becomes progressively more specific.

The Mini-specifications contained in Appendix B are a structured english description of the functions conducted by the DFD processes. Although the code generated du ing programming will look different, the mini-specification and the structure chart will clarify the programming requirements.

The Data Dictionary contained in Appendix C holds the definitions of the data stores. All data flows are subsets of the data stores. The Data Dictionary contains:

* a narrative description of each data store
* a listing of the data elements, their type and length
* the expected number of records in that data store
* valid values for data in the data store
* frequency of expected user access to the data
* identification of the key field
* which DFD modules use the data
* what squadron personnel are expected to have primary responsibility for the data correctness

The Structure Hierarchy Chart (Appendix D) displays the modules of the Structure Chart (Appendix E) as a hierarchical listing. The Structure Charts are a graphical representation of the physical design of the system. Since dBase III Plus provides the control and data definition for the system, the Structure Charts graphically depict the levels of menus used to access the functional modules. Only a few of the lowest modules show traditional passing of data between modules.

Representative outputs (Appendix F) display desired formats the Flight Schedule, Op Event Flow, Temporal Ops Schedule, Long Range Training Plan and Flight Hours Utilization Graph. All printer outputs are designed to be produced on eight-and-a-half inch by eleven inch paper.

C. DSS APPLICATION TO THIS SYSTEM

Training optimization provides the greatest benefit to the Operations and Training Department personnel. During periods of increasing operational commitments and decreasing resources, optimum use of available flights to train the average 120 crewmembers in a P-3 squadron is mandatory. Careful study of each crew position's training syllabus combined with the expertise of the instructors, Training and Operations Departments personnel should develop a decision table of training events that can be conducted simultaneously on one flight. The crewmember data base should be searched for personnel who have unfilled training requirements. This crewmember training requirements list should be evaluated to determine which personnel have the highest priority for their training. This priority training list should then be evaluated against available training opportunities, the decision table of training events, training constraints (eg., readiness requirements) and constraints imposed by management.

The DSS should allow the user:

* to select personnel training priorities manually
* to generate and optimize this data automatically
* to optimize the data generated by the user
* to allow the user to modify the data generated automatically by the DSS

These four methods would provide great flexibility to the user.

A DSS should increase the effectiveness and the efficiency of the decision making process. Effectiveness is the determination of what training activities should be considered. By evaluating the many possible training decisions, the DSS increases the user's confidence that a particular training schedule is appropriate. Efficiency implies minimizing the effort required to generate the training schedule. A DSS should investigate more training alternatives and conduct more sophisticated alternative analysis than the decision maker can without the DSS.

The DSS design should help the user conceptualize the problem by providing him appropriate graphs and tables. The DSS design should support the identification and selection of the best training alternatives. It should provide memory aids to the decision maker, be available in modes of operation which support a variety of user skills and knowledge, and it should allow the user to exercise direct personal control over the decision making process. [Ref. 7:pp. 21-28]

IV. CONCLUSIONS AND RECOMMENDATIONS

A. SYSTEM SUMMARY

The P-3 Scheduling Support System design is documented by this thesis. It was designed to be implemented using dBase III Plus on a Zenith Z-248 micro computer. A data driven structured analysis of the current manual system provided the basis from which the P-3 Scheduling Support System was designed. An interactive menu-driven system to:

* Maintain an online training record for each individual crewmember in a P-3 squadron.

* Maintain an online training record for each crew in a P-3 squadron.

* Maintain an online record of future operational commitments (Temporal Ops Data).

* Maintain an online record of future training requirements (Long Range Training Plan Data).

* Maintain an online record of Flight Hours Flown.

* Determine which personnel have unsatisfied training requirements.

* Determine which personnel are available for scheduling.

* Recording scheduled personnel as non-available.

* Automatically generate a flight schedule.

* Generate an operational event flow for flight scheduling.

* Generate a graphical comparison of allocated flight hours versus flight hours flown.

This MIS design can be used as the functional and design specifications for programming, maintenance and future enhancements of the P-3 Scheduling Support System. Additionally, with minimal modification this MIS functional specification can be used to develop analogous flight schedule generating systems for other aviation communities. This design provides a framework upon which to base a future enhanced Decision Support System. Developing a more complex DSS that would contain the basic automated MIS, additional optimization, new user interfaces and operator controls is feasible but a much more robust and higher risk project.

B. WHY STRUCTURED ANALYSIS AND DESIGN?

The primary difficulty with much of the user developed software the author has seen in the fleet, is that it is poorly planned, poorly implemented, poorly documented and maintainable only by the person who initially programmed it. Much of this software lacks the ability to respond to changing user needs.
Small software development projects, such as the P-3 Scheduling Support System, need the same lifecycle management as is used for large ADP acquisitions.[3]

[3] Office of Management and Budget Circular A-109 "Major Systems Acquisition", 1976

Mission need determination identifies and validates a need for the development of the software. Rapid response to changing high tempo operations is extremely difficult using the present manual flight scheduling systems. During mission need determination, preliminary assumptions were made and constraints were determined. During concept exploration, alternative solutions to the mission needs were developed and evaluated. The current flight scheduling system was analyzed and the functional requirements were transformed into logical specifications describing an automated flight scheduling system.

C. WHERE DO WE GO FROM HERE?

If Commander, Naval Air Force U.S. Pacific Fleet determines that the system is financially feasible, development should continue through implementation, testing and maintenance. System Development includes programming, integration, testing and validation of the operable information system to satisfy the design specification and mission needs. Deployment of an information system involves operating the system in the environment for which it was originally designed. Once in the operational environment, the user gains increased understanding of the MIS capabilities. After delivery and acceptance by the customer, the software enters the maintenance stage. Errors that escaped the testing phase require correction. New capabilities for the system are incorporated under product improvement.

Configuration management defines the documentation, control, implementation, accounting and audit of changes to the software. The delivered software is the baseline software. A formal procedure should be implemented to:

* Gather proposed changes to the baseline
* Approve the proposed changes
* Implement approved changes
* Test
* Document the changes
* Deliver the NEW BASELINE to the fleet.

One of the primary specifications for fleet applications should be user friendliness and a comprehensive user manual. Initial user training for the novice and intermediate capabilities of the hardware are also necessary. Failure to read documentation will probably never be overcome, whether it is provided in a book or online in the software. But if the user has a problem with the application, he must have the capability to answer it by reading the documentation provided.

The analyst, programmer and system acquisition manager need to realize that microcomputers represent a quantum leap from the electric typewriter and grease board that most of the aviation squadrons schedule their operations with. There is a significant percentage of enlisted personnel and officers who have no background or previous experience with computers. They don't understand them and don't like anything they don't understand. Historically, with good intentions, systems have been delivered to these same personnel without adequate training on their use. New hardware and software applications should be accompanied with immediate training.

D. RECOMMENDATIONS

Development of a DSS for the initial system is not recommended at this time. Many of the P-3 squadrons currently conduct the entire scheduling process manually. The personnel who generate flight schedules have little or no experience using computers. They could be overwhelmed by a sophisticated optimizing DSS or Expert Support System. The most important characteristics of a new MIS are usefulness and usability. Operations and Training Department personnel should first learn to use and understand the capabilities of the automated MIS. The basic MIS should undergo a period of perfective maintenance, during which time the programs are fine tuned to provide more usefulness to the users. A P-3 Scheduling Decision Support System should be instituted as a future preplanned product improvement, once the P-3

Scheduling Support System has proven its worth as an MIS.

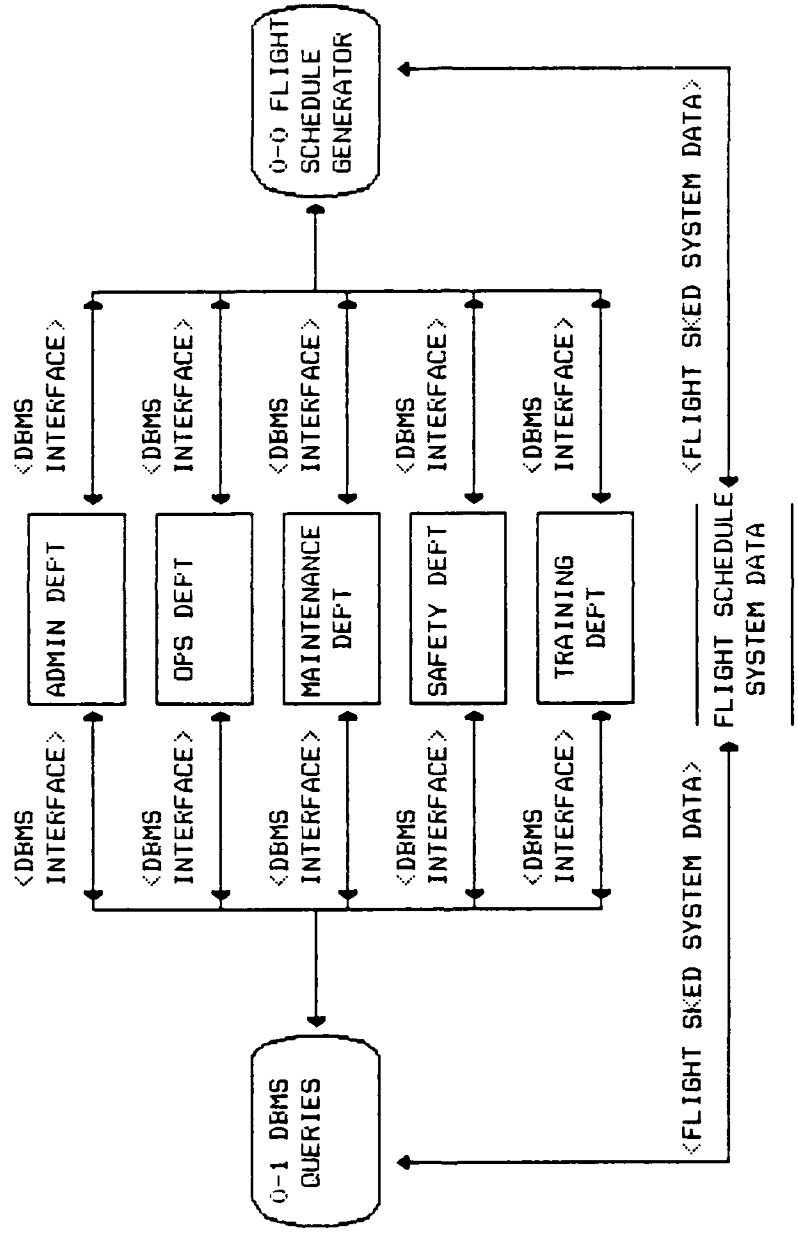
Consideration should be given to developing a recommended computer information policy for the squadrons. The introduction of such a powerful tool as the Z-248 microcomputer into a manual environment should be accompanied with guidance on the efficient use of the resource. With only two micro computers available to all departments, a squadron level ADP division should be instituted. It should be initially staffed with high capability enlisted personnel from all the user departments. A collateral duty ADP Officer might be assigned to institute the strategic guidance of the Commanding Officer regarding the priorities of micro computer resource use. The duties of the ADP officer should include archiving, backup, configuration control and integration of new resources. He should be responsible for implementing command recommendations for product improvement and acquisition to the appropriate agencies.

APPENDIX A: DATA FLOW DIAGRAMS

TABLE OF CONTENTS

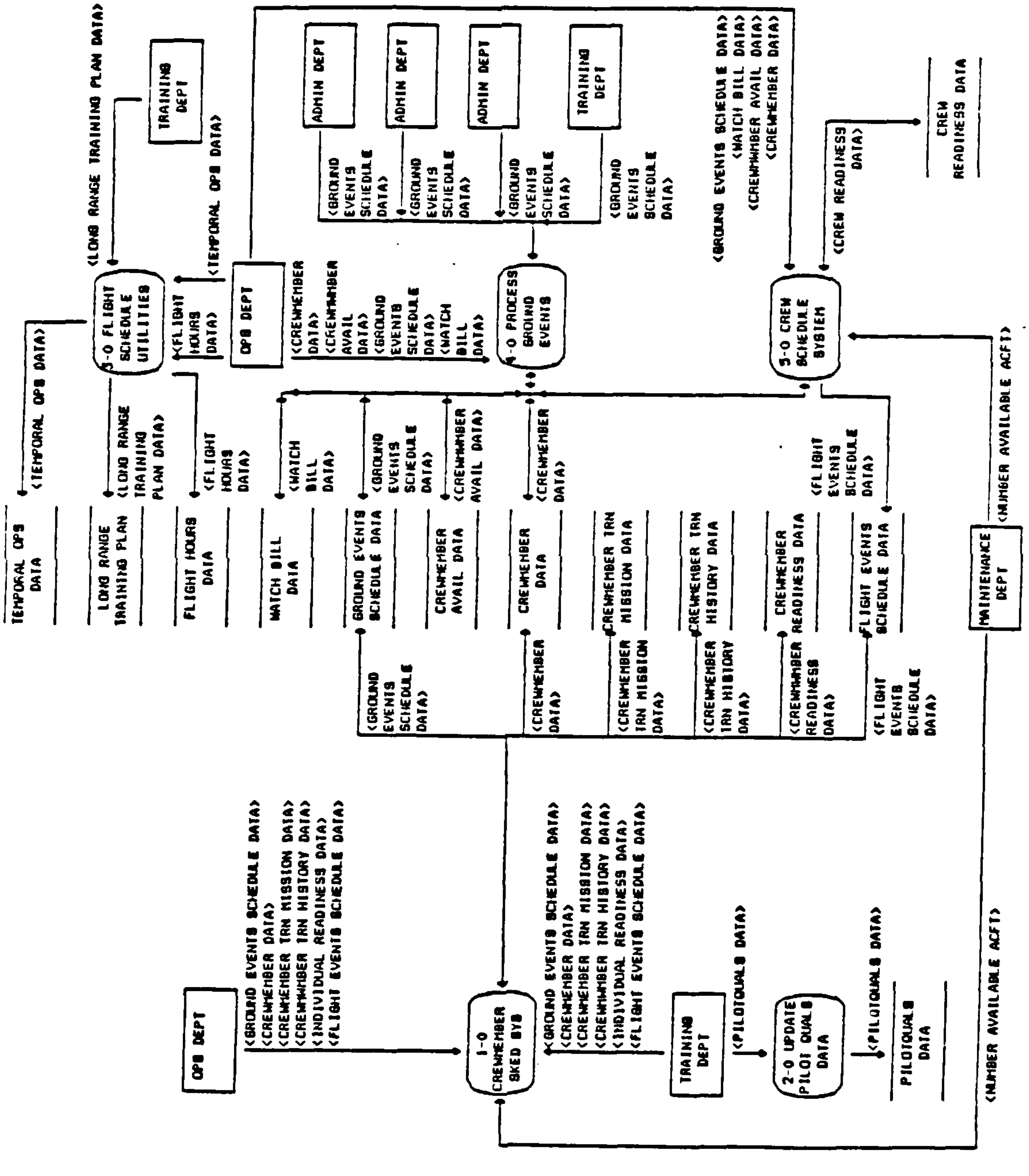
APPENDIX A: DATA FLOW DIAGRAMS

CONTEXT DIAGRAM



APPENDIX A: DATA FLOW DIAGRAMS

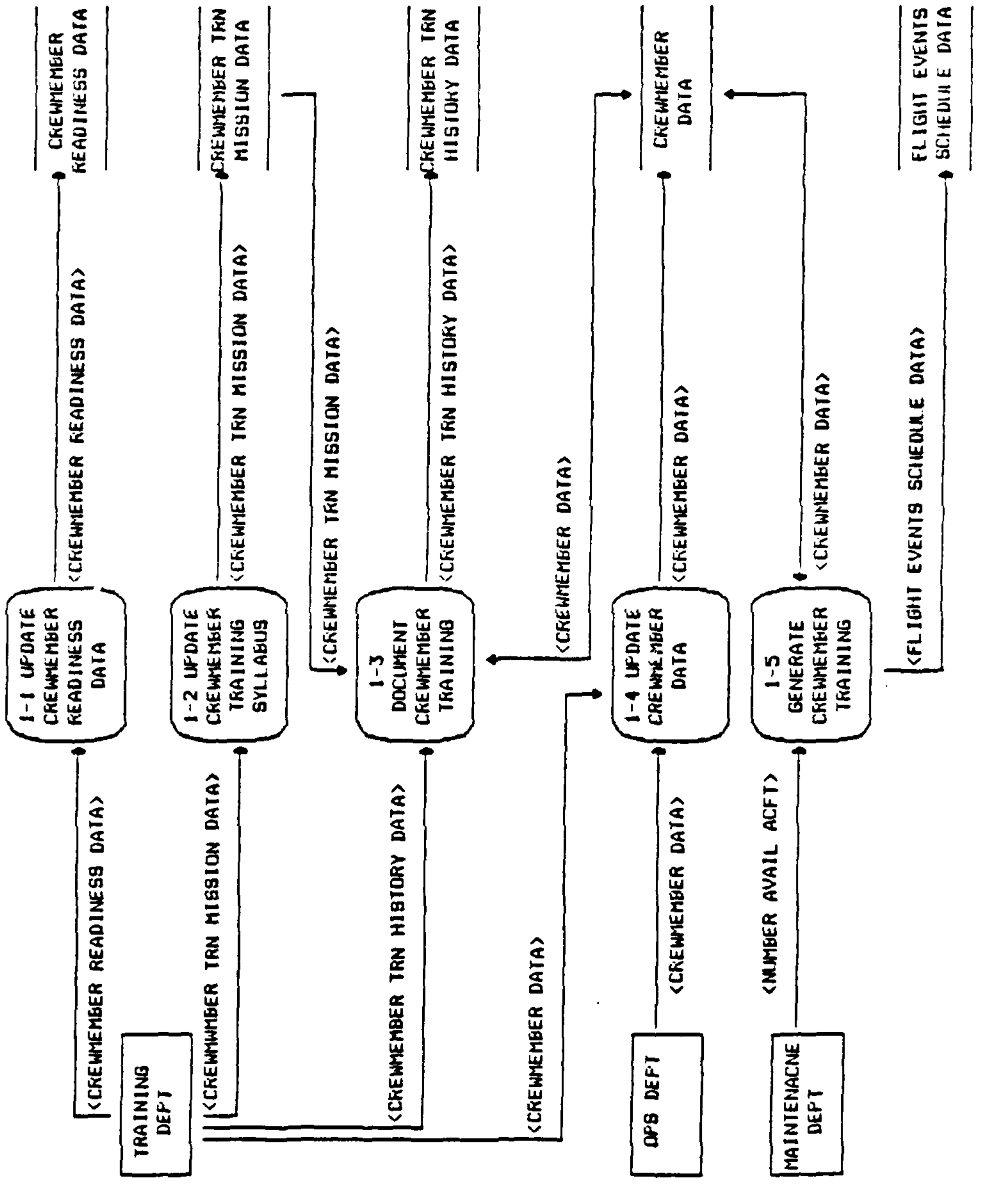
LEVEL 1



<LONG RANGE TRAINING PLAN DATA>
TRAINING DEPT
<TEMPORAL OPS DATA>
3-0 FLIGHT SCHEDULE UTILITIES
<FLIGHT HOURS DATA>
OPS DEPT
<CREWMEMBER DATA>
<CREWMMBER AVAIL DATA>
<GROUND EVENTS SCHEDULE DATA>
<WATCH BILL DATA>
ADMIN DEPT
ADMIN DEPT
ADMIN DEPT
TRAINING DEPT
<GROUND EVENTS SCHEDULE DATA>
4-0 PROCESS GROUND EVENTS
<GROUND EVENTS SCHEDULE DATA>
<WATCH BILL DATA>
<CREWMMBER AVAIL DATA>
<CREWMEMBER DATA>
<CREW READINESS DATA>
CREW READINESS DATA
5-0 CREW SCHEDULE SYSTEM
<NUMBER AVAILABLE ACFT>
<TEMPORAL OPS DATA>
<LONG RANGE TRAINING PLAN DATA>
<FLIGHT HOURS DATA>
<WATCH BILL DATA>
<GROUND EVENTS SCHEDULE DATA>
<CREWMMBER AVAIL DATA>
<CREWMEMBER DATA>
<FLIGHT EVENTS SCHEDULE DATA>
<NUMBER AVAILABLE ACFT>
TEMPORAL OPS DATA
LONG RANGE TRAINING PLAN
FLIGHT HOURS DATA
WATCH BILL DATA
GROUND EVENTS SCHEDULE DATA
CREWMEMBER AVAIL DATA
CREWMEMBER DATA
CREWMEMBER TRN MISSION DATA
CREWMEMBER TRN HISTORY DATA
CREWMEMBER READINESS DATA
FLIGHT EVENTS SCHEDULE DATA
MAINTENANCE DEPT
<GROUND EVENTS SCHEDULE DATA>
<CREWMEMBER DATA>
<CREWMEMBER TRN MISSION DATA>
<CREWMEMBER TRN HISTORY DATA>
<CREWMMBER READINESS DATA>
<FLIGHT EVENTS SCHEDULE DATA>
OPS DEPT
<GROUND EVENTS SCHEDULE DATA>
<CREWMEMBER DATA>
<CREWMMBER TRN MISSION DATA>
<CREWMMBER TRN HISTORY DATA>
<INDIVIDUAL READINESS DATA>
<FLIGHT EVENTS SCHEDULE DATA>
1-0 CREWMEMBER SKED SYS
<GROUND EVENTS SCHEDULE DATA>
<CREWMEMBER DATA>
<CREWMEMBER TRN MISSION DATA>
<CREWMMBER TRN HISTORY DATA>
<INDIVIDUAL READINESS DATA>
<FLIGHT EVENTS SCHEDULE DATA>
TRAINING DEPT
<PILOTQUALS DATA>
2-0 UPDATE PILOT QUALS DATA
<PILOTQUALS DATA>
PILOTQUALS DATA
<NUMBER AVAILABLE ACFT>

APPENDIX A: DATA FLOW DIAGRAMS

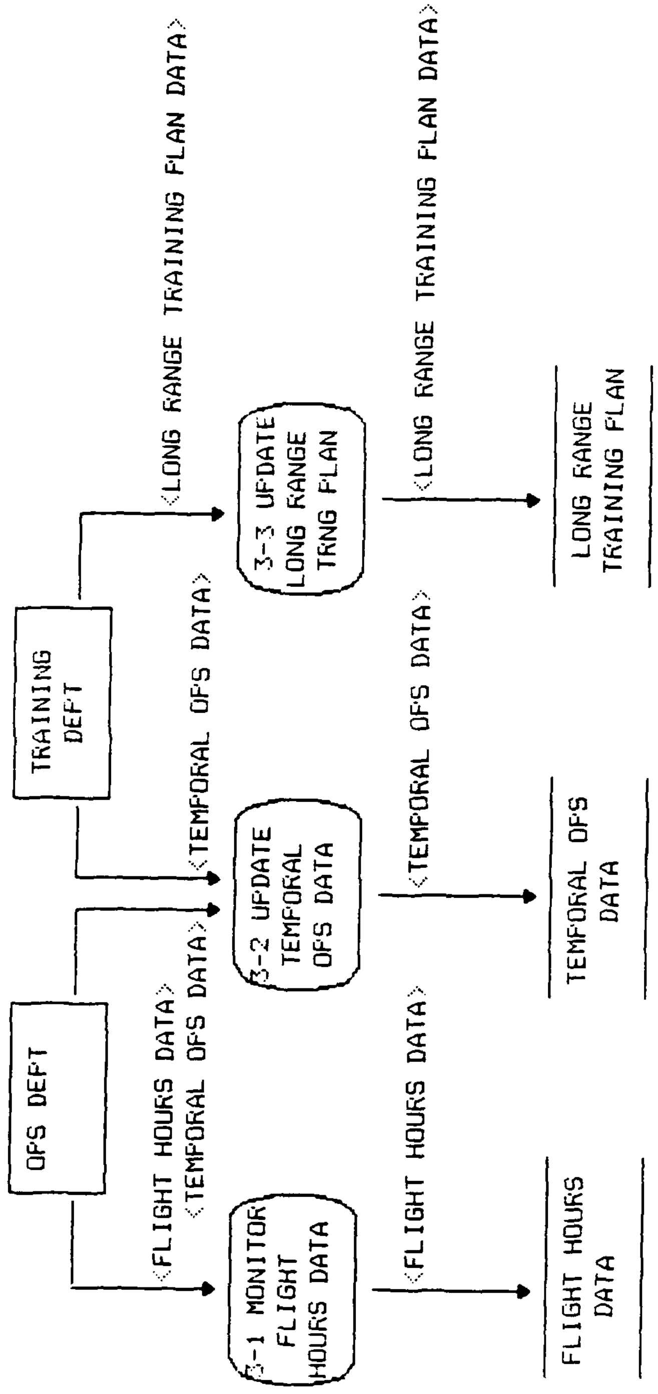
LEVEL 2 (CREWMEMBER)



CREWMEMBER READINESS DATA
CREWMEMBER TRN MISSION DATA
CREWMEMBER TRN HISTORY DATA
CREWMEMBER DATA
FLIGHT EVENTS SCHEDULE DATA
<CREWMEMBER READINESS DATA>
<CREWMEMBER TRN MISSION DATA>
<CREWMEMBER TRN MISSION DATA>
<CREWMEMBER TRN HISTORY DATA>
<CREWMEMBER DATA>
<CREWMEMBER DATA>
<CREWMEMBER DATA>
<FLIGHT EVENTS SCHEDULE DATA>
1-1 UPDATE CREWMEMBER READINESS DATA
1-2 UPDATE CREWMEMBER TRAINING SYLLABUS
1-3 DOCUMENT CREWMEMBER TRAINING
1-4 UPDATE CREWMEMBER DATA
1-5 GENERATE CREWMEMBER TRAINING
<CREWMEMBER READINESS DATA>
<CREWMWMBER TRN MISSION DATA>
<CREWMEMBER TRN HISTORY DATA>
<CREWMEMBER DATA>
<CREWMEMBER DATA>
<NUMBER AVAIL ACFT>
TRAINING DEPT
OPS DEPT
MAINTENACNE DEPT

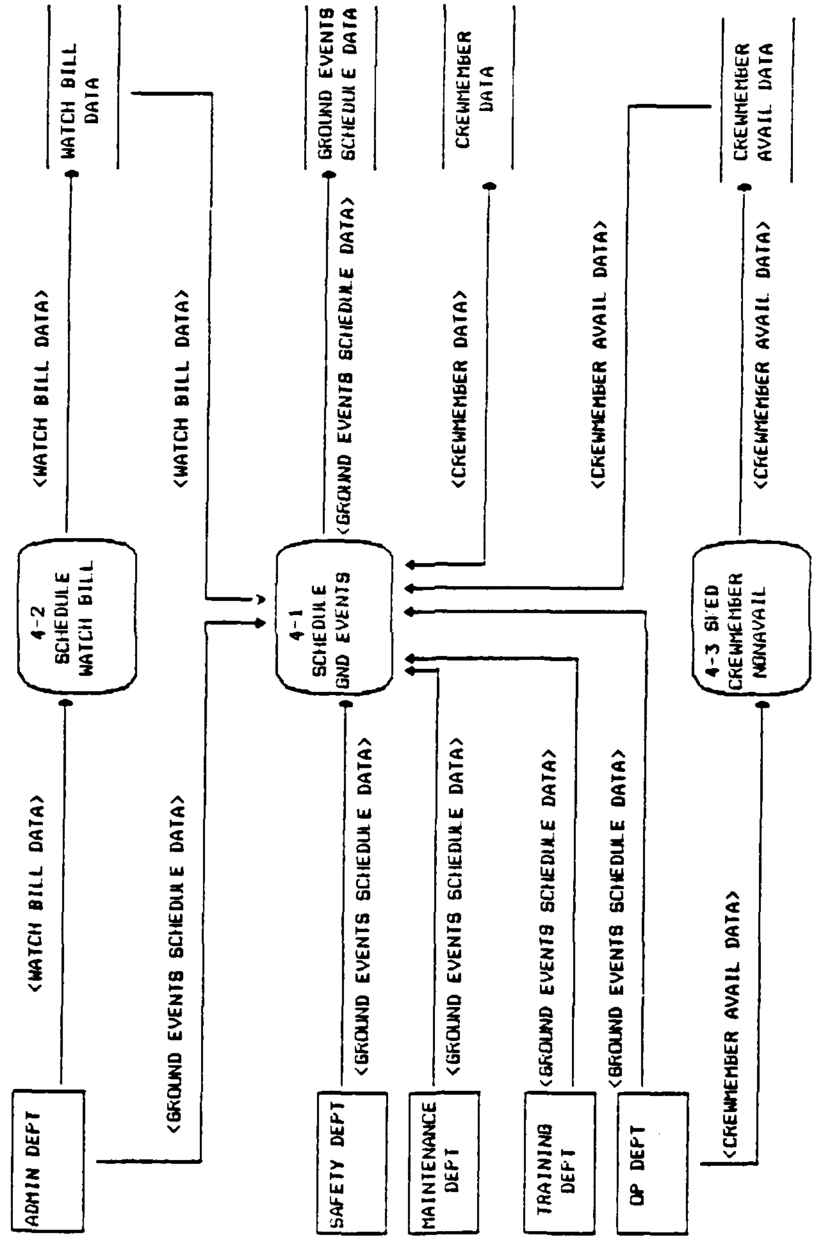
APPENDIX A: DATA FLOW DIAGRAMS

LEVEL 2 (UTILITIES)



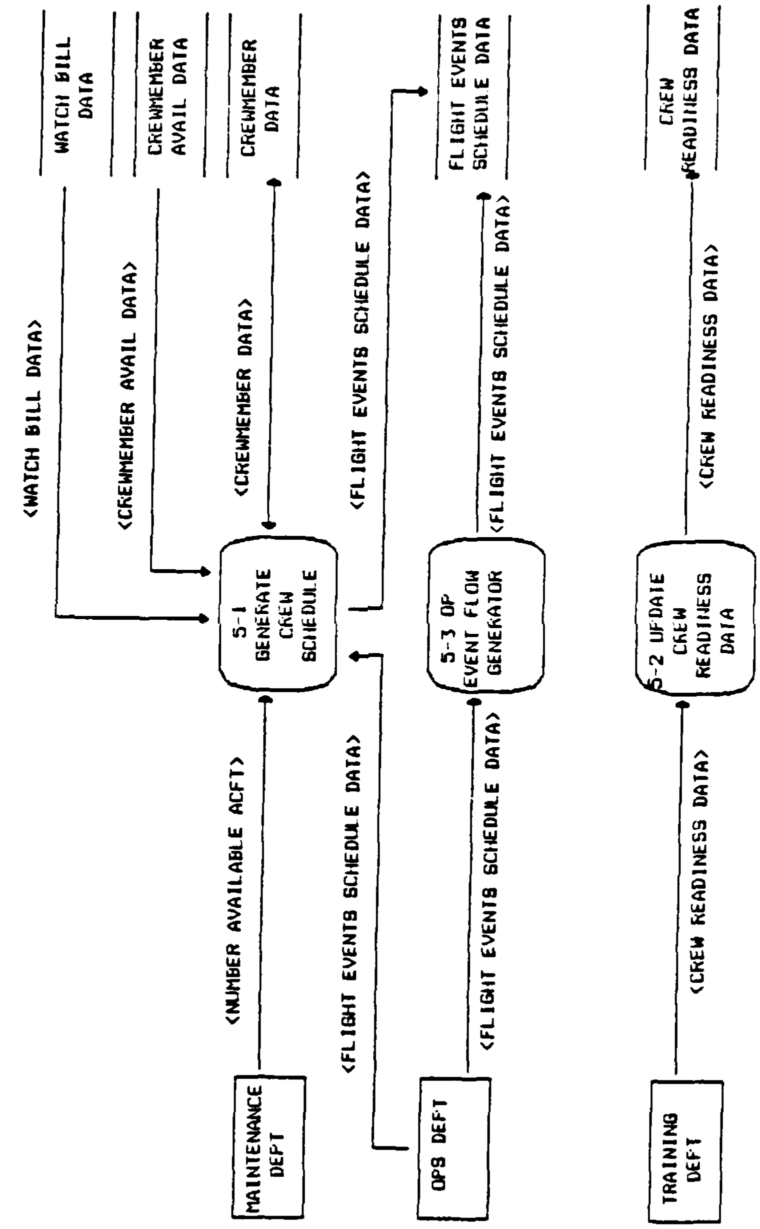
APPENDIX A: DATA FLOW DIAGRAMS

LEVEL 2 (GROUND EVENTS)



APPENDIX A: DATA FLOW DIAGRAMS

LEVEL 2 (CREW)



APPENDIX B: MINI SPECIFICATIONS

TABLE OF CONTENTS

MINI-SPECIFICATIONS

```
-----------------------------------------------------

1-1 UPDATE CREWMEMBER READINESS DATA

1 Display the following menu:
  "1. Display CREWMEMBER names.
   2. Update CREWMEMBER readiness records.
   3. Return to prior menu."

2 Case choice of '1', for each NAME in CREWMEMBER
    DATA display NAME.

3 Case choice of '2':
  3.1 Get CREWMEMBER's NAME from user,
  3.2 Find record in CREWMEMBER READINESS DATA
    where CREWMEMBER NAME = NAME.
  3.3 Enter edit mode
  3.4 Record editing changes
  3.5 Ask user to validate changes
  3.6 If changes valid, save changes made

4 Case choice of '3' return to prior menu.

-----------------------------------------------------
```

MINI-SPECIFICATIONS

```
-------------------------------------------------------

1-2  UPDATE CREWMEMBER TRAINING SYLLABUS

1 Display the following menu:
  "1. Display training syllabus MISSIONIDs.
   2. Add MISSIONID records.
   3. Change MISSIONID records.
   4. Delete MISSIONID records.
   5. Return to prior menu."

2 Case choice of ‘1’, for each MISSION in CREWMEMBER
  TRN MISSION DATA display MISSIONID.

3 Case choi e of ‘2’, enter append mode of DBMS.

4 Case choice of ‘3’:
  4.1 Get MISSIONID from user,
  4.2 Find MISSIONID’s record in CREWMEMBER TRN MISSION
      DATA
  4.3 Enter edit mode
  4.4 Record editing changes
  4.5 Ask user to validate changes
  4.6 If changes valid, save changes made

5 Case choice of ‘4’,
  5.1 Get MISSIONID from user,
  5.2 Find MISSIONID’s record in CREWMEMBER TRN MISSION
      DATA
  5.3 Display that MISSIONID record to user
  5.4 Have user verify that this record is to be
      deleted
  5.5 If user verifies record is to be removed, delete
      record

6 Case choice of ‘5 return to prior menu.
-------------------------------------------------------
```

MINI-SPECIFICATIONS

```
-------------------------------------------------------

1-3 DOCUMENT CREWMEMBER TRAINING

1 Display the following menu:
  "1. Display CREWMEMBER names.
   2. Update CREWMEMBER TRAINING HISTORY DATA.
   3. Return to prior menu."

2 Case choice of '1', for each NAME in CREWMEMBER DATA
  display NAME.

3 Case choice of '2':
  3.1 Get CREWMEMBER's NAME from user
  3.2 Get completed MISSIONID of completed training
  3.3 Get DATE MISSIONID training was accomplished
  3.4 Have user validate his entries
  3.5 If entries are validated (else goto 3.1)
    3.5.1 Find record in CREWMEMBER TRN HISTORY DATA
      where CREWMEMBER NAME = NAME.
    3.5.2 For fieldname = MISSIONID enter DATE
    3.5.3 Find MISSIONID record in CREWMEMBER TRN
      MISSION DATA.
      3.5.3.1 Skip one record
      3.5.3.1 Get new MISSIONID (next training event)
    3.5.4 Write new MISSIONID to NXTRAFLY in CREWMEMBER
      DATA

4 Case choice of '3' return to prior menu.

-------------------------------------------------------
```

MINI-SPECIFICATIONS

```
-----------------------------------------------------------
1-4 UPDATE CREWMEMBER DATA

1 Display the following menu:
  "1 Display CREWMEMBER names.
   2 Add CREWMEMBER records.
   3 Change CREWMEMBER records.
   4 Delete CREWMEMBER records.
   5 Return to prior menu."

2 Case choice of '1' for each NAME in CREWMEMBER DATA
  display NAME.

3 Case choice of '2' enter append mode of DBMS.
  3.1 If CREWPOSIT = PPC or PP2P or PP3P or PPNP
    append record to PILOTQUALS DATA where new NAME in
    PILOTQUALS DATA equals new NAME in CREWMEMBER DATA.

4 Case choice of '3':
  4.1 Get CREWMEMBER NAME from user.
  4.2 Find CREWMEMBER's record in CREWMEMBER DATA.
  4.3 Enter Edit mode of DBMS.
  4.4 Ask user to validate changes.
  4.5 If changes valid, save changes made.

5 Case choice of '4':
  5.1 Get NAME of CREWMEMBER to delete.
  5.2 Display NAME's CREWMEMBER DATA to user, have
    user verify this record is to be deleted.
  5.3 If user validates is to be removed, delete the
    record. If CREWPOSIT = PPC or PP2P or PP3P or PPNP,
    delete NAME's record from PILOTQUALS DATA.

6 Case choice of '5' return to prior menu.
-----------------------------------------------------------
```

MINI-SPECIFICATIONS

```
----------------------------------------------------
1-5 GENERATE CREWMEMBER TRAINING

1 Get NUMBER AVAIL ACFT from user.

2 Repeat until all records with instantiated NXTRAFLY
  evaluated:

  2.1 For all NAME in 3.1.1 determine how many days
    NAME has been in squadron.
  2.2 Find MISSIONID in CREWMEMBER TRN MISSION DATA
    such that MISSIONID = NXTRAFLY CREWMEMBER DATA, get
    MAXTIME.
  2.3 DATEREPORT (of CREWMEMBER DATA) minus DATE() =
    DAYSINSQUADRON.
  2.4 DAYSINSQUADRON - MAXTIME = DIFFERENCEDAYS.

  2.5 Display NAME + NXTRAFLY of records in order
    of most positive to most negative DIFFERENCEDAYS.

3 Get user choices of NAME to schedule.

4 Enter user choices in FLIGHT EVENTS SCHEDULE DATA

5 Have user verify flight schedule is correct.

6 If verified:

  6.1 AVAILABLE = LAND time (FLIGHT EVENTS SCHEDULE
    DATA) + 10 hours.

  6.2 Enter AVAILABLE in NAMEs' records in CREWMEMBER
    DATA.
----------------------------------------------------
```

MINI-SPECIFICATIONS

```
-----------------------------------------------------
2-0 UPDATE PILOT QUALS

1 Display the following menu:
  "1. Display PILOT names.
   2. Update PILOT qualification records.
   3. Return to prior menu."

2 Case choice of '1', for all NAME in PILOTQUALS DATA,
  display NAME.

3 Case choice of '2':
  3.1 Get NAME from user
  3.2 Find NAME's record in INDIVIDUAL READINESS DATA
  3.3 Enter edit mode of DBMS.
  3.4 Have user validate changes.
    3.4.1 If validated, save changes to NAME's record.

4 Case choice of '3' return to prior menu.
-----------------------------------------------------
```

```
-----------------------------------------------------
3-1 MONITOR FLIGHT HOURS DATA

1 Get HOURSFLOWN

2 Get DATEFLOWN

3 Get QUARTERHOURS

4 Get MONTH1, MONTH1DAYS,
      MONTH2, MONTH2DAYS,
      MONTH3, MONTH3DAYS.

5 Enter HOURSFLOWN and DATEFLOWN in FLIGHT HOURS DATA.

6 HOURSPERMONTH = QUARTERHOURS / 3.

7 Graph HOURSFLOWN for DATEFLOWN (where DATEFLOWN in
   MONTH1 or MONTH2 or MONTH3) versus FLIGHTHOURS *
  (nth day of quarter / days in quarter)
-----------------------------------------------------
```

MINI-SPECIFICATIONS

```
-----------------------------------------------------
3-2 UPDATE TEMPORAL OPS DATA

1 Display the following menu:
  "1. Display Temporal Ops Events
   2. Add Temporal Ops Events
   3. Delete Temporal Ops Events
   4. Change Temporal Ops Events
   5. Return to prior menu"

2 Case choice of '1':
  2.1 For all records in TEMPORAL OPS DATA,
    display EVENT, BEGIN DATE, BEGIN TIME, END DATE
    END TIME.

3 Case choice of '2', append a record to
  TEMPORAL OPS DATA, enter edit mode, have user
  validate new data.

4 Case choice of '3', get EVENT from user,
  have user verify intent to delete, delete verified
  record.

5 Case choice of '4', get EVENT from user, enter edit
  mode, have user validate changes, save validated
  changes.

6. Case choice of '5', return to prior menu.
-----------------------------------------------------
```

MINI-SPECIFICATIONS

```
-------------------------------------------------------
3-3 UPDATE LONG RANGE TRAINING PLAN

1 Display the following menu:
  "1. Display Long Range Training Events
   2. Add Long Range Training Events
   3. Delete Long Range Training Events
   4. Change Long Range Training Events
   5. Return to prior menu"

2 Case choice of '1':
  2.1 For all records in LONG RANGE TRAINING DATA,
    display EVENT, BEGIN DATE, BEGIN TIME, END DATE,
    END TIME.

3 Case choice of '2', append a record to
  LONG RANGE TRAINING DATA, enter edit mode, have user
  validate new data.

4 Case choice of '3', get EVENT from user,
  have user verify intent to delete, delete verified
  record.

5 Case choice of '4', get EVENT from user, enter edit
  mode, have user validate changes, save validated
  changes.

6 Case choice of '5', return to prior menu.
-------------------------------------------------------
-------------------------------------------------------
4-1 SCHEDULE GROUND EVENTS

1 Get from user EVENT to schedule.

2 Get from user which NAMEs to schedule in EVENT

3 Get from user BEGIN and END times

4 If NAME's AVAILABLE (CREWMEMBER DATA) <= BEGIN, enter
  user choices in GROUND EVENTS SCHEDULE DATA
  Else tell user that NAME is not available for
  scheduling.

5 Set CREWMEMBER DATA's AVAILABLE attribute equal to
  END in GROUND EVENTS SCHEDULE DATA.
-------------------------------------------------------
```

MINI-SPECIFICATIONS

4-2 SCHEDULE WATCHBILL

1 Display the following menu:
  "1 Create a WATCHBILL
   2 Return to prior menu"

2 Get WATCH to schedule from user.

3 Get NAME to schedule for the WATCH.

4 Get BEGIN time of the WATCH.

5 Get END time of the WATCH.

6 Enter NAME, BEGIN, END AND WATCH in WATCHBILL EVENTS SCHEDULE DATA.

7 Set CREWMEMBER DATA's AVAILABLE attribute equal to END in WATCHBILL EVENTS SCHEDULE DATA.

8 Get names of nonaircrew personnel to schedule for watches.

9 Enter nonaircrew watch data in WATCHBILL EVENTS SCHEDULE DATA

4-3 SCHEDULE CREWMEMBERMEMBER NONAVAIL

1 Display the following menu:
  "1 Add CREWMEMBER to NONAVAILABILITY LIST.
   2 Delete CREWMEMBER from NONAVAILABILITY LIST.
   3 Update CREWMEMBER NONAVAILABILITY.
   4 Return to prior menu."
2 Case choice of '1', append a new record on CREWMEMBER AVAIL.
3 Case choice of '2', list all NAMEs in the file and get NAME to delete. Display Name's record. Have user intent to delete this record. If validated, Delete NAME's record.
4 Case choice of '3', list all NAMEs in the file and get NAME to update. Enter DBMS edit mode. Have user verify changes prior to saving.
5 Case choice of '4', return to prior menu.

MINI-SPECIFICATIONS

```
5-1 GENERATE CREW SKED

1 Display following menu:
  "1 Update CREW readiness.
   2 Generate an OP EVENT FLOW.
   3 Generate CREW schedules.
   4 Return to prior menu."

2 Case choice of '1', call Module 5-2 UPDATE CREW
READINESS DATA.

3 Case choice of '2', call module 5-3 OP EVENT FLOW
  GENERATOR.

4 Case choice of '3',:
  4.1 Get NUMBER AVAIL ACFT from user.
    4.2 Get CREWNUMBER to schedule for flight.
      4.2.1 For all NAME in CREWMEMBER, where CREW =
        CREWNUMBER, enter NAME in FLIGHT EVENTS
        SCHEDULE DATA.
        4.2.1.1 Enter Edit mode.
        4.2.1.2 Have user verify entries.
        4.2.1.3 If verified, save entries.
      4.2.2 Enter AVAILABLE in CREWMEMBER DATA.
        AVAILABLE = LAND (FLIGHT EVENTS SCHEDULE DATA)
        + 10 hours.

5 Case choice of '4', return to prior menu.
```

APPENDIX C: DATA DICTIONARY

TABLE OF CONTENTS

APPENDIX C: DATA DICTIONARY

Identification:
Name: CREW READINESS DATA
Alias: none
Narrative Description: Crew Readiness Data represents the historic accomplishment of readiness qualifications by the crews.

Representation:

Field Name	Type	Width
Crew	Integer	2
A30	Date	8
A32	Date	8
A32AVAIL	Real	6
A32ACH	Real	6
A34	Date	8
A34AVAIL	Real	6
A34ACH	Real	6
A35	Date	8
A35AVAIL	Real	6
A35ACH	Real	6
A36	Date	8
A37	Date	8
A38	Date	8
A39	Date	8
OSAP	Date	8
ASWOTP	Date	8
WSTOTP	Date	8

Expected number of records: less than 150
Frequency of User Access: weekly
Relationships:
Key Field(s): Crew
Used in: Modules 5-1 (Generate Crew Sked)
Primary User: Readiness Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
Name: CREWMEMBER AVAIL DATA
Narrative Description: The Crewmember Avail Data represents crewmembers who are not available for flights or ground event scheduling during certain times due to administrative reasons (illness, Temporary Additional Duty etc.)

Representation:

Field Name	Type	Width
*NAME	Character	15
EVENT	Character	10
MONTHBGN	Numeric	2
MONTHEND	Numeric	2
BEGIN	Numeric (DTG)	6
END	Numeric (DTG)	6

Expected number of records: less than 20

Valid Values:
DATE BEGIN & DATE END -- 0 to 12
TIME BEGIN & TIME END --
100001 - 010059, ... 012300 - 012359
to
310001 - 310059, ... 312300 - 312359
(first two digits = day of the month
last four digits = hours and minutes in
a 24 hour clock)

Frequency of User Access: daily

Relationships:
Key Field(s): NAME
Used In: Module 4-3 (SCHEDULE CREWMEMBER NONAVAIL)
Module 4-1 (SCHEDULE GROUND EVENTS)
Module 5-1 (GENERATE CREW SCHEDULE)

Primary User: Schedules Officer
Secondary User:

APPENDIX C: DATA DICTIONARY

Identification:

Name: CREWMEMBER DATA

Alias:

Narrative Description: Crewmember Data represent general personnel information and important qualifications and dates

Representation:

Field Name	Type	Width
*NAME	Character	15
AVAILABLE	Integer (DTG)	6
NXTRAFLY	Character	10
CREW	Integer	2
HOURS30	Real	5
TOTALHRS	Real	7
BIRTHDAY	Date	8
LASTPHYSIC	Date	8
UPCHIT	Logical	1
DATEREPORT	Date	8
NATOPSCHECK	Date	8
LPNV	Date	8
DWEST	Date	8
BTCREWMAN	Logical	1
ISARCREW	Logical	1
INSTRUCTOR	Logical	1
BLUECARD	Logical	1
RANK	Character	4
CREWPOSIT	Character	3

Expected number of records: less than 150

Frequency of User Access: Daily

Relationships:

Key Field(s): NAME

Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),
Module 4-1 (SCHEDULE GROUND EVENTS)
Module 5-1 (GENERATE CREW SCHEDULE)

Primary User: Schedules Officer

Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:

Name: CREWMEMBER TRN MISSION DATA

Alias: NFO TRN MISSION DATA, PILOT TRN MISSION DATA, FE TRN MISSION DATA, IFT TRN MISSION DATA, SS3 TRN MISSION DATA, SS1/2 TRN MISSION DATA, RDO TRN MISSION DATA, 2MECH TRN MISSION DATA, ORD TRN MISSION DATA

Narrative Description: CREWMEMBER TRNG MISSION DATA is a list of required PQS flights for each aircrew position. Additionally, the number of days from reporting aboard the squadron to when the flight should be completed is stored.

Representation:

Field Name	Type	Width
*MISSIONID	Character	10
MAXTIME	Numeric	3

Expected number of records: less than 400

Valid Values:

MAXTIME -- 0 to 548 (18 months)

Frequency of User Access: Rarely Changes

Relationships:

Key Field(s): MISSION ID

Used In: Module 1-2 (UPDATE AIRCREW TRN SYLLABUS)

Primary User: Training Officer

Secondary User:

APPENDIX C: DATA DICTIONARY

Identification:
  Name: FLIGHT ENGINEER TRNG HISTORY DATA
  Alias: CREWMEMBER TRN HISTORY DATA
  Narrative Description: Fe Trng History Data is a historical record of the date each PQS flight is accomplished by each FE enroute his positional qualification

Representation:

Field Name	Type	Width
*NAME	Character	15
FE_OFT1	Date	8
FE_FLY1	Date	8
FE_FLY2	Date	8
FE_OFT2	Date	8
FE_FLY3	Date	8
FE_FLY4	Date	8
FE_FLY5	Date	8
FE_FLY6	Date	8
FE_FLY7	Date	8
FE_OFT3	Date	8
FE_FLY8	Date	8
FE_FLY9	Date	8
FE_TAC1	Date	8
FE_TAC2	Date	8
FENATOPS	Date	8

Expected number of records: less than 150
Frequency of User Access: Daily
Relationships:
  Key Field(s): NAME
  Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),
Primary User: Schedules Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
  Name: FLIGHT EVENTS SCHEDULE DATA
  Alias: none
  Narrative Description: The Flight Events Schedule Data represent the flight events section of a flight schedule.

Representation:

Field Name	Type	Width
*EVENTNUM	Integer	2
PREFLIGHT	Integer (DTG)	6
TAKEOFF	Integer (DTG)	6
DURATION	Real	5
LAND	Integer (DTG)	6
CREW	Integer	2
ADDNAME1	Character	20
ADDNAME2	Character	20
ADDNAME3	Character	20
ADDNAME4	Character	20
ADDNAME5	Character	20
DELNAME1	Character	20
DELNAME2	Character	20
DELNAME3	Character	20
DELNAME4	Character	20
DELNAME5	Character	20
FLYCODE	Character	3
TYPEFLT	Character	6
NOTENUM1	Integer	1
NOTENUM2	Integer	1

Expected number of records: less than 20

Valid Values:
  PREFLIGHT & TAKEOFF & LAND --
    100001 - 010059, ... 012300 - 012359
      to
    310001 - 310059, ... 312300 - 312359
    (first two digits - day of the month
     last four digits - hours and minutes in
     a 24 hour clock)

Frequency of User Access: Daily

Relationships:
  Key Field(s): Event Number
  Used In: Module 5-1 (GENERATE CREW SCHEDULE)

Primary User: Schedules Officer

Secondary User:

APPENDIX C: DATA DICTIONARY

Identification:
Name: FLIGHT HOURS DATA
Alias: None
Narrative Description: FLIGHT HOURS DATA is maintained to compute the number of flight hours remaining from those allotted by higher authority. Flight hours used each day are entered to enable graphical trend analysis of the flight hour utilization.

Representation:

Field Name	Type	Width
*DATEFLOWN	Date	8
HOURSFLOWN	Numeric	3

Expected number of records: less than 100
Frequency of User Access: Daily update
Relationships:
Key Field(s): DATE
Used In: Module 3-1 (MONITOR FLIGHT HOURS DATA)
Primary User: Operations Officer
Secondary User: None

APPENDIX C: DATA DICTIONARY

Identification:
Name: GROUND EVENTS SCHEDULE DATA
Alias:
Narrative Description: Ground Events Schedule Data represents the daily Administrative, Training, Maintenance and Safety events.

Representation:

Field Name	Type	Width
*NAME	Character	20
RANK	Character	4
EVENT	Character	20
BEGIN	Numeric (DTG)	6
END	Numeric (DTG)	6
PLACE	Character	8
NOTENUMS	Character	8

Expected number of records: less than 20
Valid Values:
BEGIN & END --
100001 - 010059, ... 012300 - 012359
to
310001 - 310059, ... 312300 - 312359
(first two digits = day of the month
last four digits = hours and minutes in a 24 hour clock)
Frequency of User Access: daily
Relationships:
Key Field(s): NAME
Used In: Module 4-1 (SCHEDULE GROUND EVENTS)
Primary User: Schedules Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:

Name: INDIVIDUAL READINESS DATA

Alias: none

Narrative Description: Individual Readiness Data represents the historic accomplishment of Readiness Qualifications by the individual Crewmembers.

Representation:

Field Name	Type	Width
*NAME	Character	15
A3	Date	8
A4	Date	8
A5	Date	8
A6	Date	8
A7	Date	8
A8	Date	8
A9	Date	8
A10	Date	8
A12	Date	8
A13	Date	8
A14	Date	8
A15	Date	8
A20	Date	8
A21	Date	8
A30	Date	8
A31	Date	8
A32	Date	8
A34	Date	8
A35	Date	8
A36	Date	8
A37	Date	8
A38	Date	8
A39	Date	8

Expected number of records: less than 150

Frequency of User Access: weekly

Relationships:

Key Field(s): NAME

Used In: Modules 1-1 (UPDATE AIRCREW READINESS)

Primary User: Readiness Officer

Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:

Name: INFLIGHT TECHNICIAN TRNG HISTORY DATA

Alias: CREWMEMBER TRN HISTORY DATA

Narrative Description: IFT Trng History Data is a historical record of the date each PQS flight is accomplished by each IFT enroute his positional qualification

Representation:

Field Name	Type	Width
*NAME	Character	15
OBS_FLY1	Date	8
OBS_FLY2	Date	8
OBS_FLY3	Date	8
OBS_APU	Date	8
OBS_WING	Date	8
OBSNATOPS	Date	8
IFT_GND1	Date	8
IFT_GND2	Date	8
IFT_GND3	Date	8
IFT_FLY1	Date	8
IFT_FLY2	Date	8
IFT_FLY3	Date	8
IFTNATOPS	Date	8

Expected number of records: less than 150

Frequency of User Access: Daily

Relationships:

Key Field(s): NAME

Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),

Primary User: Schedules Officer

Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:

Name: LONG RANGE TRAINING PLAN

Alias: none

Narrative Description: The Long Range Training Plan contains the schools and associated training for all squadron personnel for up to a year.

Representation:

Field Name	Type	Width
*NAME	Character	20
EVENT	Character	20
MONTHBGN	Integer	2
BEGIN	Integer (DTG)	6
MONTHEND	Integer	2
END	Integer	6

Expected number of records: less than 100

Valid Values:

BEGIN TIME 0001 - 0059, 0100 - 0159 ... 2300 - 2359

END TIME 0001 - 0059, 0100 - 0159 ... 2300 - 2359

Frequency of User Access: Ad Hoc

Relationships:

Key Field(s): NAME

Used In: Module and 3-3 (UPDATE LONG RANGE TRAINING PLAN)

Primary User: Training Officer

Secondary User: Schedules Officer

APPENDIX C: DATA DICTIONARY

Identification:
Name: MECH TRNG HISTORY DATA
Alias: CREWMEMBER TRN HISTORY DATA
Narrative Description: MECH TRNG HISTORY DATA is a historical record of the date each PQS flight is accomplished by each MECH enroute his positional qualification

Representation:

Field Name	Type	Width
*NAME	Character	15
MECHOFT1	Date	8
MECHFLY1	Date	8
MECHFLY2	Date	8
MECHOFT2	Date	8
MECHFLY3	Date	8
MECHFLY4	Date	8
MECHFLY5	Date	8
MECHFLY6	Date	8
MECHFLY7	Date	8
MECHOFT3	Date	8
MECHFLY8	Date	8
MECHFLY9	Date	8
MECHTAC1	Date	8
MECHTAC2	Date	8
MENATOPS	Date	8

Expected number of records: less than 150
Frequency of User Access: Daily
Relationships:
Key Field(s): NAME
Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),
Primary User: Schedules Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
Name: NFO TRNG HISTORY DATA
Alias: CREWMEMBER TRN HISTORY DATA
Narrative Description: NFO Trng History Data is a historical record of the date each PQS flight is accomplished by each NFO enroute his positional qualification

Representation:

Field Name	Type	Width
*NAME	Character	15
NAV1	Date	8
NAV2	Date	8
NAV3	Date	8
NAV4	Date	8
NAV5	Date	8
NAV6	Date	8
NAV7	Date	8
NAV8	Date	8
NAV9	Date	8
NAVNATOPS	Date	8
TAC1	Date	8
TAC2	Date	8
TAC3	Date	8
TAC4	Date	8
TAC5	Date	8
TAC6	Date	8
TAC7	Date	8
TAC8	Date	8
TAC9	Date	8
TACNATOPS	Date	8

Expected number of records: less than 150
Frequency of User Access: Daily
Relationships:
Key Field(s): NAME
Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),
Primary User: Schedules Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
  Name: NOTES
  Alias:
  Narrative Description: Notes are Flight schedule and ground schedule notes which are placed at the bottom of the flight schedule to reduce redundancy and save space. It is logically part of the Flight Schedule Event Data and Ground Schedule Event Data, but seperated to reduce redundancy.

Representation:

Field Name	Type	Width
*NOTES	Character	80

Expected number of records: less than 10

Frequency of User Access: Daily

Relationships:
  Key Field(s): NOTE
  Used In: Modules 5-1 (GENERATE CREW SCHEDULE) and 4-1 (SCHEDULE GROUND EVENTS)

Primary User: Skedules Officer

Secondary User: Operations Officer

APPENDIX C: DATA DICTIONARY

Identification:

Name: ORDNANCEMAN TRNG HISTORY DATA

Alias: CREWMEMBER TRN HISTORY DATA

Narrative Description: ORD Trng History Data is a historical record of the date each PQS flight is accomplished by each ORD enroute his positional qualification

Representation:

Field Name	Type	Width
*NAME	Character	15
OBS_FLY1	Date	8
OBS_FLY2	Date	8
OBS_FLY3	Date	8
OBS_APU	Date	8
OBS_WING	Date	8
OBSNATOPS	Date	8
ORD_FLY1	Date	8
ORD_FLY2	Date	8
ORD_FLY3	Date	8
ORD_FLY4	Date	8
ORD_FLY5	Date	8
ORD_FLY6	Date	8
ORD_FLY7	Date	8
ORDNATOPS	Date	8

Expected number of records: less than 150

Frequency of User Access: Daily

Relationships:

Key Field(s): NAME

Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),

Primary User: Schedules Officer

Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
Name: PILOTQUALS
Alias:
Narrative Description: Pilotquals are Pilot specific data and qualifications, such as Maintenance Check Pilot and number of landings.

Representation:

Field Name	Type	Width
*NAME	Character	15
NEWLANDING	Integer	2
NEWAPROACH	Integer	2
OLDLANDING	Integer	2
OLDAPROACH	Integer	2
WEAPSPILOT	Logical	1
COURRIER	Logical	1
MAINTENANC	Logical	1
INSTRUMENT	Logical	1

Expected number of records: less than 40
Frequency of User Access:
Relationships:
Key Field(s): NAME
Used In: Module 2-0 (UPDATE PILOT DATA)
Primary User: Skedules Officer
Secondary User: Operations Officer

APPENDIX C: DATA DICTIONARY

Identification:
  Name: PILOT TRNG HISTORY DATA
  Alias: CREWMEMBER TRN HISTORY DATA
  Narrative Description: Pilot Trng History Data is a historical record of the date each PQS flight is accomplished by each pilot enroute his positional qualification.

Representation:

Field Name	Type	Width
*NAME	Character	15
PP3PINDOC	Date	8
PP3P_OFT1	Date	8
PP3P_FLY1	Date	8
PP3P_FLY2	Date	8
PP3P_OFT2	Date	8
PP3P_FLY3	Date	8
PP3P_OFT3	Date	8
PP3P_FLY4	Date	8
PP3P_FLY5	Date	8
PP3P_FLY6	Date	8
PP3P_NAV	Date	8
PP2P_OFT2	Date	8
PP2P_FLY1	Date	8
PP2P_FLY2	Date	8
PP2P_OFT2	Date	8
PP2P_FLY3	Date	8
PP2P_OFT3	Date	8
PP2P_FLY4	Date	8
PP2P_TAC1	Date	8
PP2P_TAC2	Date	8
PP2P_WST1	Date	8
PP2P_TAC3	Date	8
PP2P_NAV	Date	8
PP2PNATOP	Date	8
PPC_OFT1	Date	8
PPC_FLY1	Date	8
PPC_FLY2	Date	8
PPC_OFT2	Date	8
PPC_FLY3	Date	8
PPC_FLY4	Date	8
PPC_OFT3	Date	8
PPC_FLY5	Date	8
PPC_WST1	Date	8
PPC_TAC1	Date	8
PPC_WST2	Date	8
PPC_TAC2	Date	8
PPC_TAC3	Date	8
PPC_WST3	Date	8

APPENDIX C: DATA DICTIONARY

PPC_TAC4	Date	8
PPC_WST4	Date	8
PPC_TAC5	Date	8
PPC_CHECK	Date	8

Expected number of records: less than 40
Frequency of User Access: Daily
Relationships:
  Key Field(s): NAME
  Used In: Module 2-3 (DOCUMENT AIRCREW TRN)
Primary User: Training Officer
Secondary User:

Identification:
  Name: RADIOMAN TRNG HISTORY DATA
  Alias: CREWMEMBER TRN HISTORY DATA
  Narrative Description: RDO Trng History Data is a historical record of the date each PQS flight is accomplished by each RDO enroute his positional qualification
Representation:

Field Name	Type	Width
*NAME	Character	15
OBS_FLY1	Date	8
OBS_FLY2	Date	8
OBS_FLY3	Date	8
OBS_APU	Date	8
OBS_WING	Date	8
OBSNATOPS	Date	8
RDO_GND1	Date	8
RDO_GND2	Date	8
RDO_GND3	Date	8
RDO_FLY1	Date	8
RDO_FLY2	Date	8
RDO_FLY3	Date	8
RDONATOPS	Date	8

Expected number of records: less than 150
Frequency of User Access: Daily
Relationships:
  Key Field(s): NAME
  Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),
Primary User: Schedules Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
Name: SS1/2 TRNG HISTORY DATA
Alias: CREWMEMBER TRN HISTORY DATA
Narrative Description: SS1/2 Trng History Data is a historical record of the date each PQS flight is accomplished by each SS2 enroute his positional qualification

Representation:

Field Name	Type	Width
*NAME	Character	15
OBS_FLY1	Date	8
OBS_FLY2	Date	8
OBS_FLY3	Date	8
OBS_APU	Date	8
OBS_WING	Date	8
OBSNATOPS	Date	8
SS2_PFT1	Date	8
SS2_WST1	Date	8
SS2_WST2	Date	8
SS2_FLY1	Date	8
SS2_WST3	Date	8
SS2_FLY2	Date	8
SS2_WST4	Date	8
SS2_WST5	Date	8
SS2_FLY3	Date	8
SS2_FLY4	Date	8
SS2NATOPS	Date	8

Expected number of records: less than 150
Frequency of User Access: Daily
Relationships:
Key Field(s): NAME
Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),
Primary User: Schedules Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
Name: SS3 TRNG HISTORY DATA
Alias: CREWMEMBER TRN HISTORY DATA
Narrative Description: SS3 Trng History Data is a historical record of the date each PQS flight is accomplished by each SS3 enroute his positional qualification

Representation:

Field Name	Type	Width
*NAME	Character	15
OBS_FLY1	Date	8
OBS_FLY2	Date	8
OBS_FLY3	Date	8
OBS_APU	Date	8
OBS_WING	Date	8
OBSNATOPS	Date	8
SS3_WST1	Date	8
SS3_WST2	Date	8
SS3_WST3	Date	8
SS3_WST4	Date	8
SS3_PFT1	Date	8
SS3_FLY1	Date	8
SS3_FLY2	Date	8
SS3_FLY3	Date	8
SS3_FLY4	Date	8
SS3_FLY5	Date	8
SS3_FLY6	Date	8
SS3NATOPS	Date	8

Expected number of records: less than 150
Frequency of User Access: Daily
Relationships:
Key Field(s): NAME
Used In: Modules 1-4 (UPDATE CREWMEMBER DATA),
Primary User: Schedules Officer
Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:

Name: TEMPORAL OPS DATA

Alias: none

Narrative Description: Temporal Ops commitments are operational commitments that reoccur on a regular basis, (eg. The Ready Alert Cycle, Hosting Duties, Corrosion Inspections). These Temporal commitments generally require advanced preparation and planning. During the Ready Alert Cycle, fewer training flights are flown, because the squadron is required to be capable of sustaining an extended ASW prosecution. TEMPORAL OPS DATA is essentially a calendar of these events.

Representation:

Field Name	Type	Width
*EVENT	Character	10
MONTHBGN	Integer	2
BEGIN	Integer	6
MONTHEND	Integer	2
END	Integer	6

Expected number of records: less than 100

Valid Values:

BEGIN 010001 - 010059 ... 312300 - 312359
END 010001 - 010059 ... 312300 - 312359

Frequency of User Access: Ad Hoc

Relationships:

Key Field(s): Event

Used In: Module 3-2 (Update Temporal Ops Data)

Primary User: Schedules Officer (Operations Department)

Secondary User: Training Officer

APPENDIX C: DATA DICTIONARY

Identification:
  Name: WATCH BILL DATA
  Alias:
  Narrative Description: The Watch Bill Data represents the scheduled watches for squadron personnel.

Representation:

Field Name	Type	Width
*NAME	Character	20
BEGIN	Integer (DTG)	6
END	Integer (DTG)	6
WATCH	Character	10

Expected number of records: less than 200

Valid Values:
  BEGIN & END --
    100001 - 010059, ... 012300 - 012359
                  to
    310001 - 310059, ... 312300 - 312359
    (first two digits = day of the month
     last four digits = hours and minutes in a 24 hour clock)

Frequency of User Access: daily

Relationships:
  Key Field(s): *NAME*
  Used In: Module 4-2 (SCHEDULE WATCH BILL)

Primary User: Schedules Officer

Secondary User: Senior Watch Officer/Command Master Chief

APPENDIX D: STRUCTURE HIERARCHY CHART

```
0.0 MAIN MENU
  1.0 CREWMEMBER SCHEDULING SYSTEM MENU
    1.1 UPDATE CREWMEMBER READINESS
      1.1.1 UPDATE NFO READINESS
        1.1.1.1 DISPLAY NFO NAMES
        1.1.1.2 UPDATE NFO READINESS RECORDS
        1.6 RETURN TO PRIOR MENU
      1.1.2 UPDATE PILOT READINESS
        1.1.2.1 DISPLAY PILOT NAMES
        1.1.2.2 UPDATE PILOT READINESS RECORDS
        1.6 RETURN TO PRIOR MENU
      1.1.3 UPDATE SS3 READINESS
        1.1.3.1 DISPLAY SS3 NAMES
        1.1.3.2 UPDATE SS3 READINESS RECORDS
        1.6 RETURN TO PRIOR MENU
      1.1.4 UPDATE SS1/2 READINESS
        1.1.4.1 DISPLAY SS1/2 NAMES
        1.1.4.2 UPDATE SS1/2 READINESS RECORDS
        1.6 RETURN TO PRIOR MENU
      1.1.5 UPDATE ORD READINESS
        1.1.5.1 DISPLAY ORD NAMES
        1.1.5.2 UPDATE ORD READINESS RECORDS
        1.6 RETURN TO PRIOR MENU
    1.2 UPDATE CREWMEMBER TRAINING SYLLABUS
      1.2.1 UPDATE NFO TRAINING SYLLABUS
        1.2.1.1 DISPLAY NFO TRAINING SYLLABUS
        1.2.1.2 ADD MISSIONID RECORDS
        1.2.1.3 CHANGE MISSIONID RECORDS
        1.2.1.4 DELETE MISSIONID RECORDS
        1.6 RETURN TO PRIOR MENU
      1.2.2 UPDATE PILOT TRAINING SYLLABUS
        1.2.1.2 DISPLAY PILOT TRAINING SYLLABUS
        1.2.1.2 ADD MISSIONID RECORDS
        1.2.1.3 CHANGE MISSIONID RECORDS
        1.2.1.4 DELETE MISSIONID RECORDS
        1.6 RETURN TO PRIOR MENU
      1.2.3 UPDATE SS3 TRAINING SYLLABUS
        1.2.3.1 DISPLAY SS3 TRAINING SYLLABUS
        1.2.1.2 ADD MISSIONID RECORDS
        1.2.1.3 CHANGE MISSIONID RECORDS
        1.2.1.4 DELETE MISSIONID RECORDS
        1.6 RETURN TO PRIOR MENU
```

APPENDIX D: STRUCTURE HIERARCHY CHART

```
  1.2.4 UPDATE SS1/2 TRAINING SYLLABUS
    1.2.4.1 DISPLAY SS1/2 TRAINING SYLLABUS
    1.2.1.2 ADD MISSIONID RECORDS
    1.2.1.3 CHANGE MISSIONID RECORDS
    1.2.1.4 DELETE MISSIONID RECORDS
    1.6 RETURN TO PRIOR MENU
  1.2.5 UPDATE ORD TRAINING SYLLABUS
    1.2.5.1 DISPLAY ORD TRAINING SYLLABUS
    1.2.1.2 ADD MISSIONID RECORDS
    1.2.1.3 CHANGE MISSIONID RECORDS
    1.2.1.4 DELETE MISSIONID RECORDS
    1.6 RETURN TO PRIOR MENU
  1.2.6 UPDATE FE TRAINING SYLLABUS
    1.2.6.1 DISPLAY FE TRAINING SYLLABUS
    1.2.1.2 ADD MISSIONID RECORDS
    1.2.1.3 CHANGE MISSIONID RECORDS
    1.2.1.4 DELETE MISSIONID RECORDS
    1.6 RETURN TO PRIOR MENU
  1.2.7 UPDATE 2MECH TRAINING SYLLABUS
    1.2.7.1 DISPLAY 2MECH TRAINING SYLLABUS
    1.2.1.2 ADD MISSIONID RECORDS
    1.2.1.3 CHANGE MISSIONID RECORDS
    1.2.1.4 DELETE MISSIONID RECORDS
    1.6 RETURN TO PRIOR MENU
  1.2.8 UPDATE IFT TRAINING SYLLABUS
    1.2.8.1 DISPLAY IFT TRAINING SYLLABUS
    1.2.1.2 ADD MISSIONID RECORDS
    1.2.1.3 CHANGE MISSIONID RECORDS
    1.2.1.4 DELETE MISSIONID RECORDS
    1.6 RETURN TO PRIOR MENU
  1.2.9 UPDATE RDO TRAINING SYLLABUS
    1.2.9.1 DISPLAY RDO TRAINING SYLLABUS
    1.2.1.2 ADD MISSIONID RECORDS
    1.2.1.3 CHANGE MISSIONID RECORDS
    1.2.1.4 DELETE MISSIONID RECORDS
    1.6 RETURN TO PRIOR MENU
1.3 DOCUMENT CREWMEMBER TRAINING
  1.3.1 DOCUMENT NFO TRAINING
    1.1.1.1 DISPLAY NFO NAMES
    1.3.1.1 UPDATE NFO TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
  1.3.2 DOCUMENT NFO TRAINING
    1.1.2.1 DISPLAY PILOT NAMES
    1.3.2.1 UPDATE PILOT TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
  1.3.3 DOCUMENT SS3 TRAINING
    1.1.3.1 DISPLAY SS3 NAMES
    1.3.3.1 UPDATE SS3 TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
```

APPENDIX D: STRUCTURE HIERARCHY CHART

```
  1.3.4 DOCUMENT SS1/2 TRAINING
    1.1.4.1 DISPLAY SS1/2 NAMES
    1.3.4.1 UPDATE SS1/2 TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
  1.3.5 DOCUMENT ORD TRAINING
    1.1.5.1 DISPLAY ORD NAMES
    1.3.5.1 UPDATE ORD TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
  1.3.6 DOCUMENT FE TRAINING
    1.3.6.1 DISPLAY FE NAMES
    1.3.6.2 UPDATE FE TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
  1.3.7 DOCUMENT 2MECH TRAINING
    1.3.7.1 DISPLAY 2MECH NAMES
    1.3.7.2 UPDATE 2MECH TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
  1.3.8 DOCUMENT IFT TRAINING
    1.3.8.1 DISPLAY IFT NAMES
    1.3.8.2 UPDATE IFT TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
  1.3.9 DOCUMENT RDO TRAINING
    1.3.9.1 DISPLAY RDO NAMES
    1.3.9.2 UPDATE RDO TRAINING HISTORY DATA
    1.6 RETURN TO PRIOR MENU
1.4 UPDATE CREWMEMEER DATA
  1.4.1 DISPLAY CREWMEMBER NAMES
    1.1.1.1 DISPLAY NFO NAMES
    1.1.2.1 DISPLAY PILOT NAMES
    1.1.3.1 DISPLAY SS3 NAMES
    1.1.4.1 DISPLAY SS1/2 NAMES
    1.1.5.1 DISPLAY ORD NAMES
    1.3.6.1 DISPLAY FE NAMES
    1.3.7.1 DISPLAY 2MECH NAMES
    1.3.8.1 DISPLAY IFT NAMES
    1.3.9.1 DISPLAY RDO NAMES
  1.4.2 ADD CREWMEMBER RECORDS
  1.4.3 CHANGE CREWMEMBER RECORDS
  1.4.4 DELETE CREWMEMBER RECORDS
  1.6 RETURN TO PRIOR MENU
1.5 GENERATE CREWMEMBER TRAINING
  1.5.1 GET NUMBER OF AVAILABLE AIRCRAFT
  1.5.2 DETERMINE HIGH PRIORITY TRAINING EVENTS
  1.5.3 GET USER TRAINING EVENT CHOICES
  1.5.4 ENTER USER CHOICES INTO FLIGHT SCHEDULE
        DATA
  1.5.5 UPDATE CREWMEMBER AVAILABILITY
1.6 RETURN TO PRIOR MENU
```

APPENDIX D: STRUCTURE HIERARCHY CHART

```
2.0 UPDATE PILOT QUALIFICATION DATA
  1.1.2.1 DISPLAY PILOT NAMES
  2.1 UPDATE PILOT QUALIFICATION RECORDS
  1.6 RETURN TO PRIOR MENU
3.0 FLIGHT SCHEDULE UTILITIES
  3.1 MONITOR FLIGHT HOURS DATA
  3.2 UPDATE TEMPORAL OPS DATA
    3.2.1 DISPLAY TEMPORAL OPS EVENTS
    3.2.2 ADD TEMPORAL OPS EVENTS
    3.2.3 CHANGE TEMPORAL OPS EVENTS
    3.2.4 DELETE TEMPORAL OPS EVENTS
    1.6 RETURN TO PRIOR MENU
  3.3 UPDATE LONG RANGE TRAINING PLAN
    3.3.1 DISPLAY LONG RANGE TRAINING PLAN
    3.3.2 ADD LONG RANGE TRAINING PLAN EVENTS
    3.3.3 CHANGE LONG RANGE TRAINING PLAN EVENTS
    3.3.4 DELETE LONG RANGE TRAINING PLAN EVENTS
    1.6 RETURN TO PRIOR MENU
  1.6 RETURN TO PRIOR MENU
4.0 PROCESS GROUND EVENTS
  4.1 SCHEDULE GROUND EVENTS
    4.1.1 DISPLAY GROUND EVENTS
    4.1.2 ADD GROUND EVENTS
    4.1.3 CHANGE GROUND EVENTS
    4.1.4 DELETE GROUND EVENTS
    1.6 RETURN TO PRIOR MENU
  4.2 SCHEDULE WATCH BILL
    4.2.1 DISPLAY WATCH BILL DATA
    4.2.2 ADD WATCH BILL DATA
    4.2.3 CHANGE WATCH BILL DATA
    4.2.4 DELETE WATCH BILL DATA
    1.6 RETURN TO PRIOR MENU
  4.3 SCHEDULE CREWMEMBER NONAVAIL
    4.3.1 DISPLAY CREWMEMBER NONAVAIL DATA
    4.3.2 ADD CREWMEMBER NONAVAIL DATA
    4.3.3 CHANGE CREWMEMBER NONAVAIL DATA
    4.3.4 DELETE CREWMEMBER NONAVAIL DATA
    1.6 RETURN TO PRIOR MENU
  1.6 RETURN TO PRIOR MENU
5.0 GENERATE CREW SCHEDULE
  5.1 UPDATE CREW READINESS
  5.2 GENERATE OP EVENT FLOW
    5.2.1 GET OP EVENT DATA FROM USER
      5.2.1.1 VALIDATE TIME
    5.2.2 CALC OP EVENT FLOW
    5.2.3 OUTPUT OP EVENT FLOW DATA
```

APPENDIX D: STRUCTURE HIERARCHY CHART

```
  5.3 CREW SCHEDULER
    1.5.1 GET NUMBER OF AVAILABLE AIRCRAFT
    5.3.1 GET CREW EVENT DATA
    5.3.2 ENTER CREW EVENT DATA INTO FLIGHT SCHEDULE
          DATA
    1.5.5 UPDATE CREWMEMBER AVAILABILITY
  1.6 RETURN TO PRIOR MENU
6.0 EXIT THE PROGRAM
```

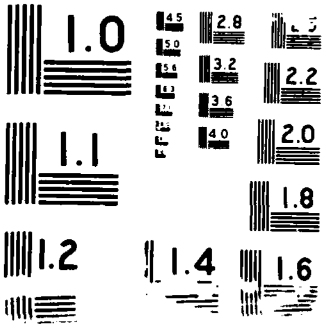
APPENDIX E: STRUCTURE CHARTS

TABLE OF CONTENTS

APPENDIX E: STRUCTURE CHARTS

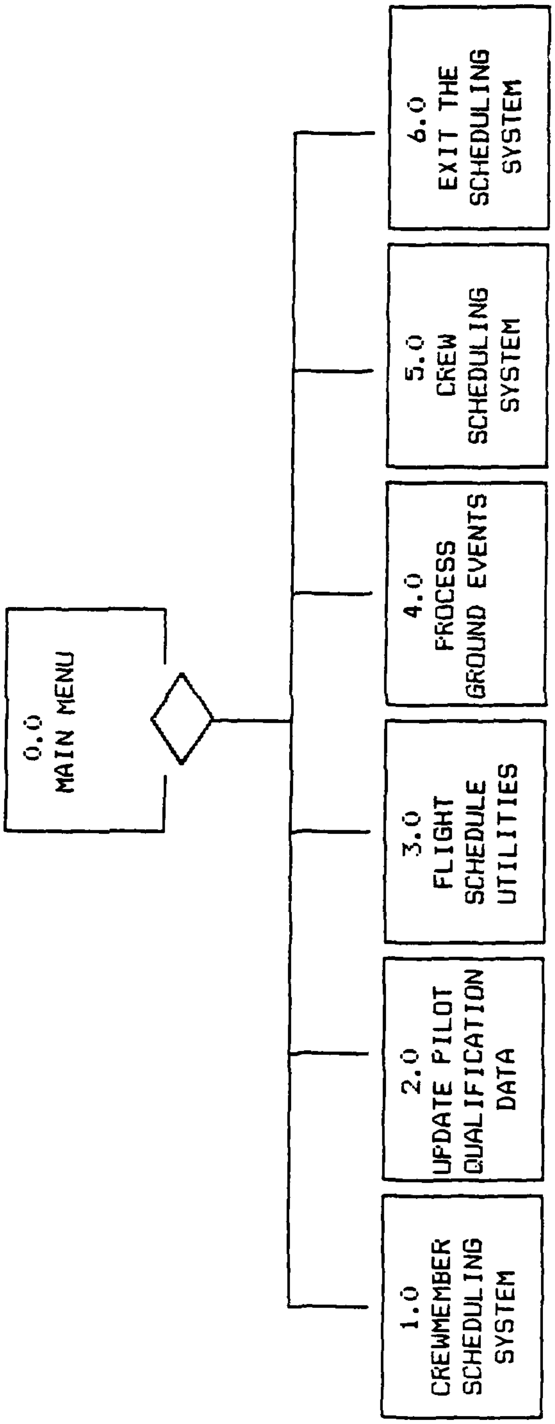
AD-A194 824 THE P-3 SCHEDULING SUPPORT SYSTEM (P-3 S3)(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W B ANDERSON MAR 88 2/2

UNCLASSIFIED F/G 1/3.4 NL


1.0
4.5
5.0
5.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.2
1.4
1.6

APPENDIX E: STRUCTURE CHARTS

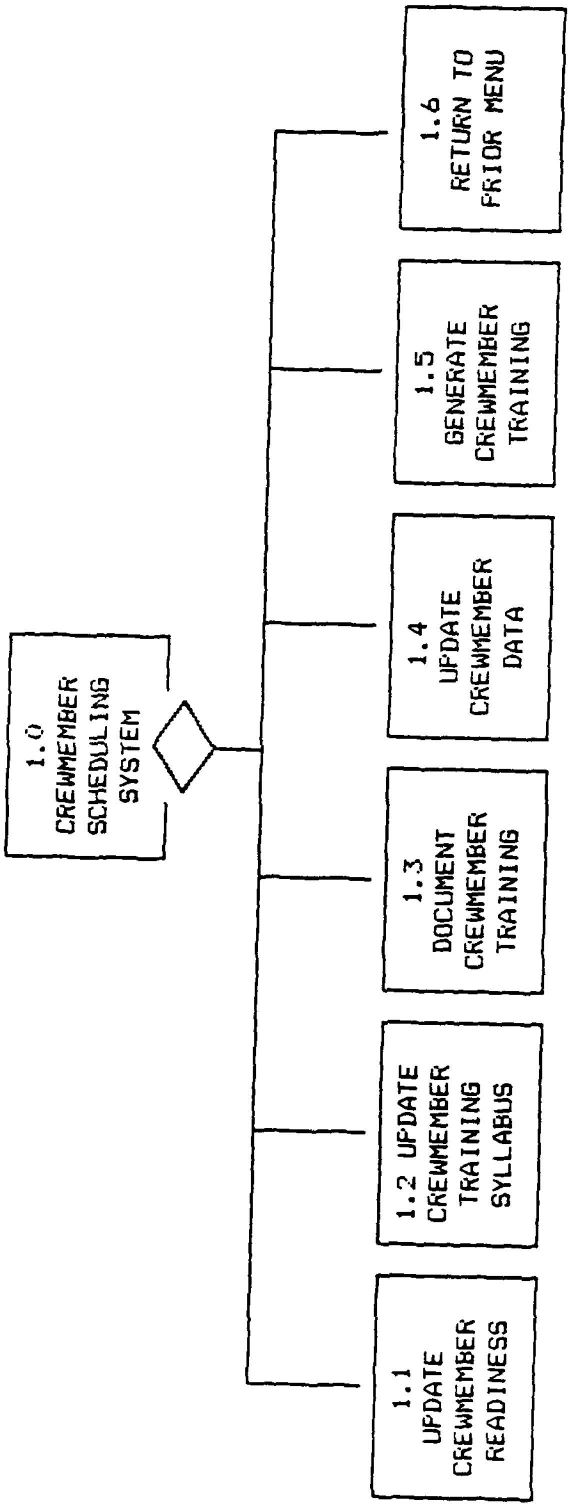
0.0 MAIN MENU



0.0 MAIN MENU
1.0 CREWMEMBER SCHEDULING SYSTEM
2.0 UPDATE PILOT QUALIFICATION DATA
3.0 FLIGHT SCHEDULE UTILITIES
4.0 PROCESS GROUND EVENTS
5.0 CREW SCHEDULING SYSTEM
6.0 EXIT THE SCHEDULING SYSTEM

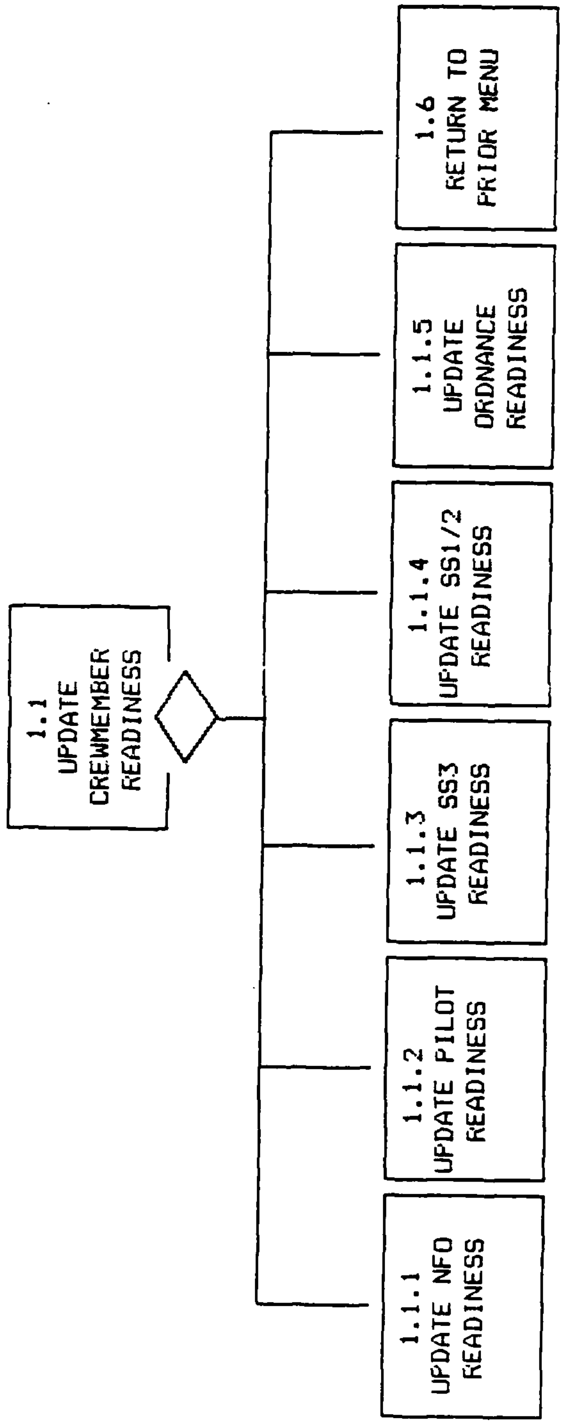
APPENDIX E: STRUCTURE CHARTS

1.0 CREWMEMBER SCHEDULING SYSTEM MENU


1.0 CREWMEMBER SCHEDULING SYSTEM
1.1 UPDATE CREWMEMBER READINESS
1.2 UPDATE CREWMEMBER TRAINING SYLLABUS
1.3 DOCUMENT CREWMEMBER TRAINING
1.4 UPDATE CREWMEMBER DATA
1.5 GENERATE CREWMEMBER TRAINING
1.6 RETURN TO PRIOR MENU

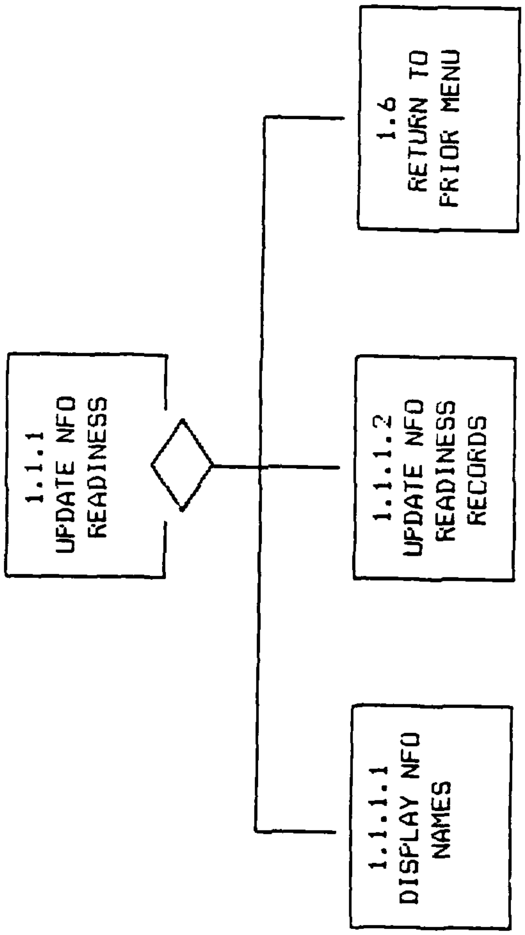
APPENDIX E: STRUCTURE CHARTS

1.1 UPDATE CREWMEMBER READINESS


1.1 UPDATE CREWMEMBER READINESS
1.1.1 UPDATE NFO READINESS
1.1.2 UPDATE PILOT READINESS
1.1.3 UPDATE SS3 READINESS
1.1.4 UPDATE SS1/2 READINESS
1.1.5 UPDATE ORDNANCE READINESS
1.6 RETURN TO PRIOR MENU

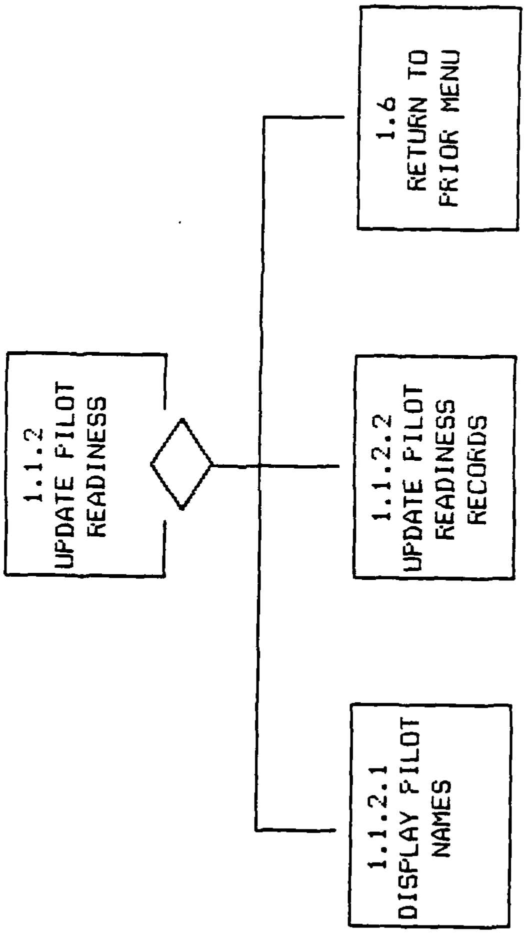
APPENDIX E: STRUCTURE CHARTS

1.1.1 UPDATE NFO READINESS


1.1.1 UPDATE NFO READINESS
1.6 RETURN TO PRIOR MENU
1.1.1.2 UPDATE NFO READINESS RECORDS
1.1.1.1 DISPLAY NFO NAMES

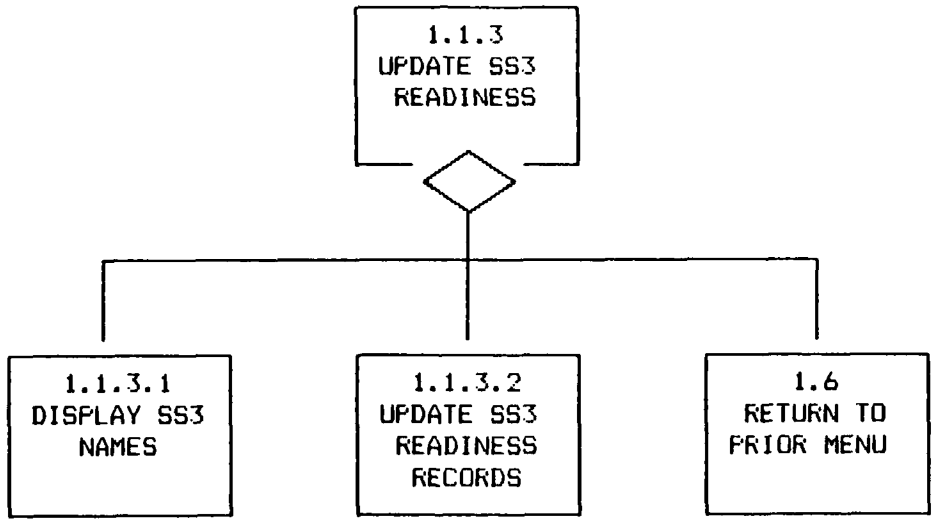
APPENDIX E: STRUCTURE CHARTS

1.1.2 UPDATE PILOT READINESS



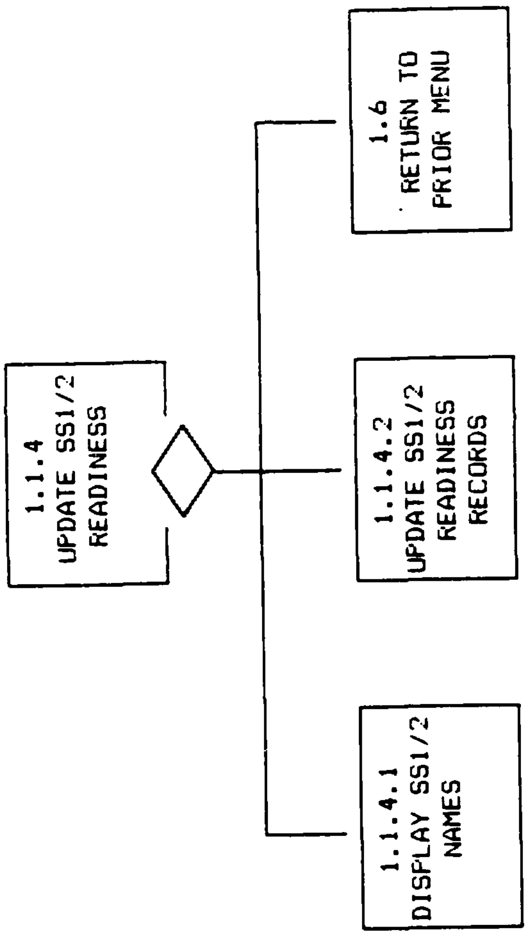
APPENDIX E: STRUCTURE CHARTS

1.1.3 UPDATE SS3 READINESS



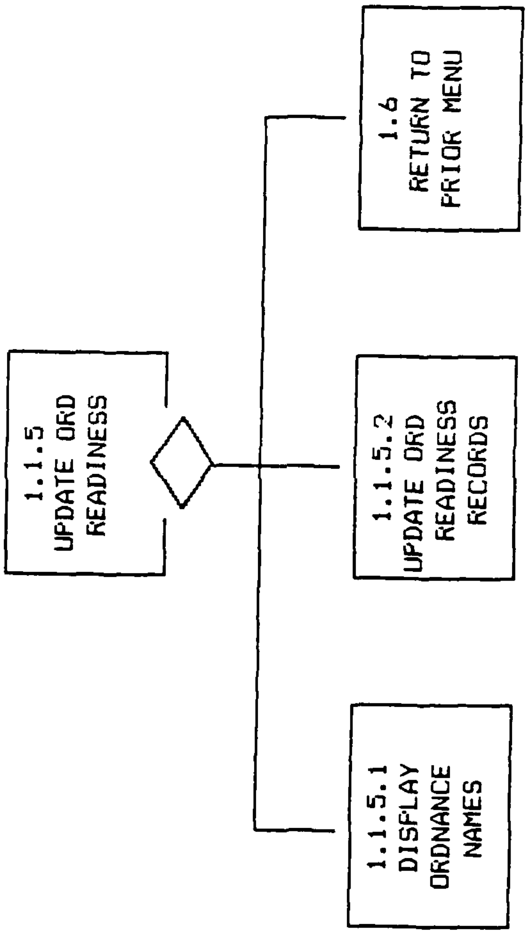
APPENDIX E: STRUCTURE CHARTS

1.1.4 UPDATE SS1/2 READINESS



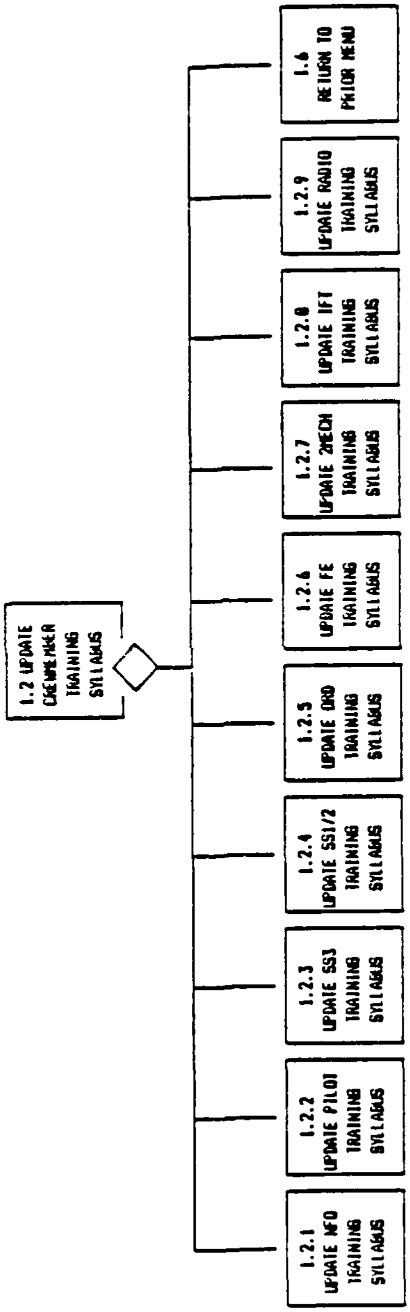
APPENDIX E: STRUCTURE CHARTS

1.1.5 UPDATE ORDNANCE READINESS



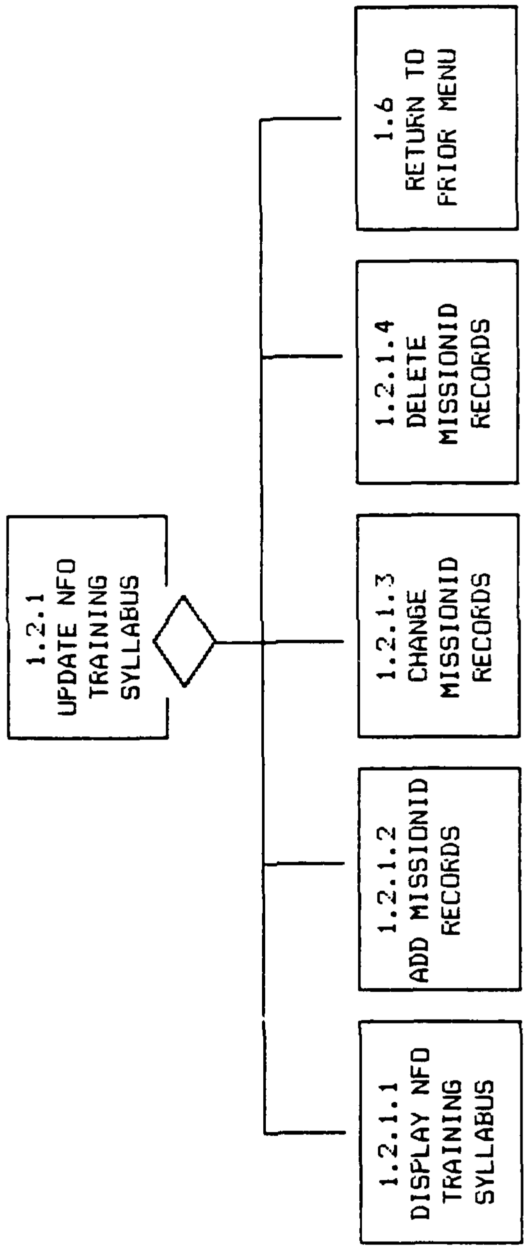
APPENDIX E: STRUCTURE CHARTS

1.2 UPDATE CREWMEMBER TRAINING SYLLABUS


1.2 UPDATE CREWMEMBER TRAINING SYLLABUS
1.2.1 UPDATE NFO TRAINING SYLLABUS
1.2.2 UPDATE PILOT TRAINING SYLLABUS
1.2.3 UPDATE SS3 TRAINING SYLLABUS
1.2.4 UPDATE SS1/2 TRAINING SYLLABUS
1.2.5 UPDATE ORD TRAINING SYLLABUS
1.2.6 UPDATE FE TRAINING SYLLABUS
1.2.7 UPDATE 2MECH TRAINING SYLLABUS
1.2.8 UPDATE IFT TRAINING SYLLABUS
1.2.9 UPDATE RADIO TRAINING SYLLABUS
1.6 RETURN TO PRIOR MENU

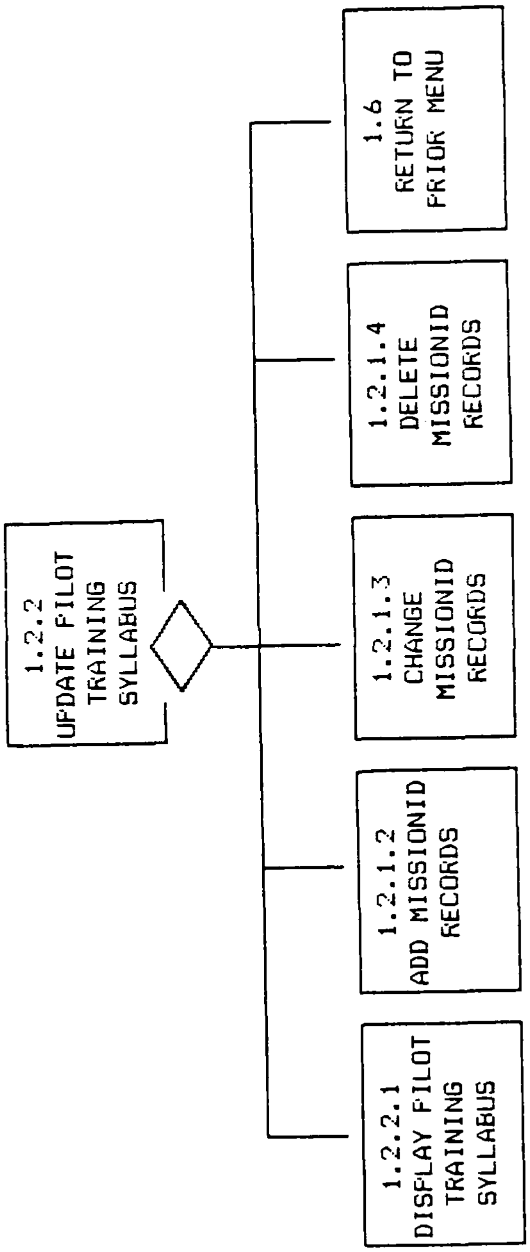
APPENDIX E: STRUCTURE CHARTS

1.2.1 UPDATE NFO TRAINING SYLLABUS



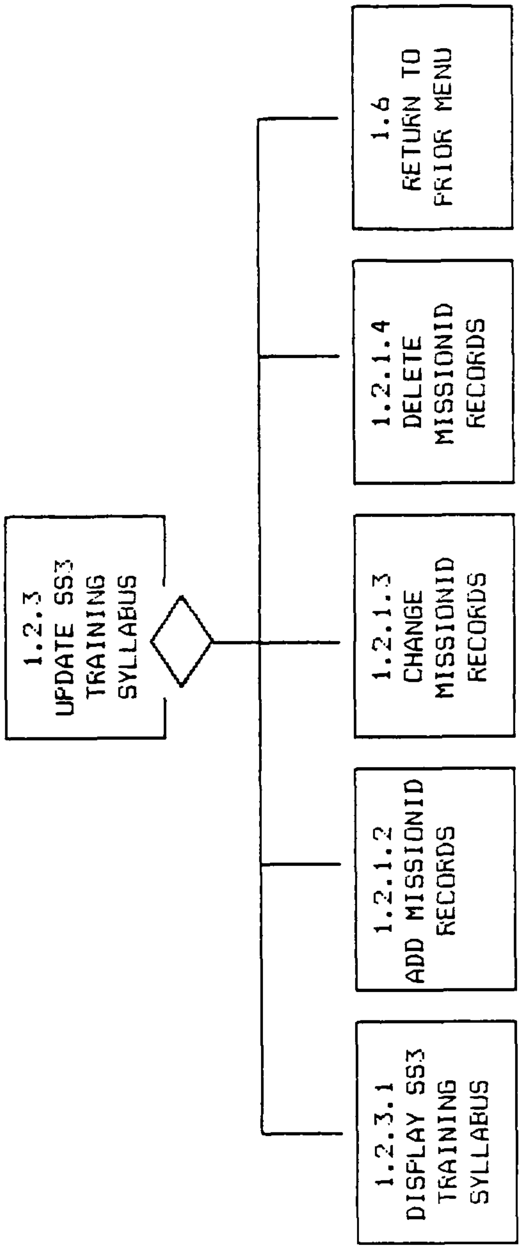
APPENDIX E: STRUCTURE CHARTS

1.2.2 UPDATE PILOT TRAINING SYLLABUS


1.2.2 UPDATE PILOT TRAINING SYLLABUS
1.2.2.1 DISPLAY PILOT TRAINING SYLLABUS
1.2.1.2 ADD MISSIONID RECORDS
1.2.1.3 CHANGE MISSIONID RECORDS
1.2.1.4 DELETE MISSIONID RECORDS
1.6 RETURN TO PRIOR MENU

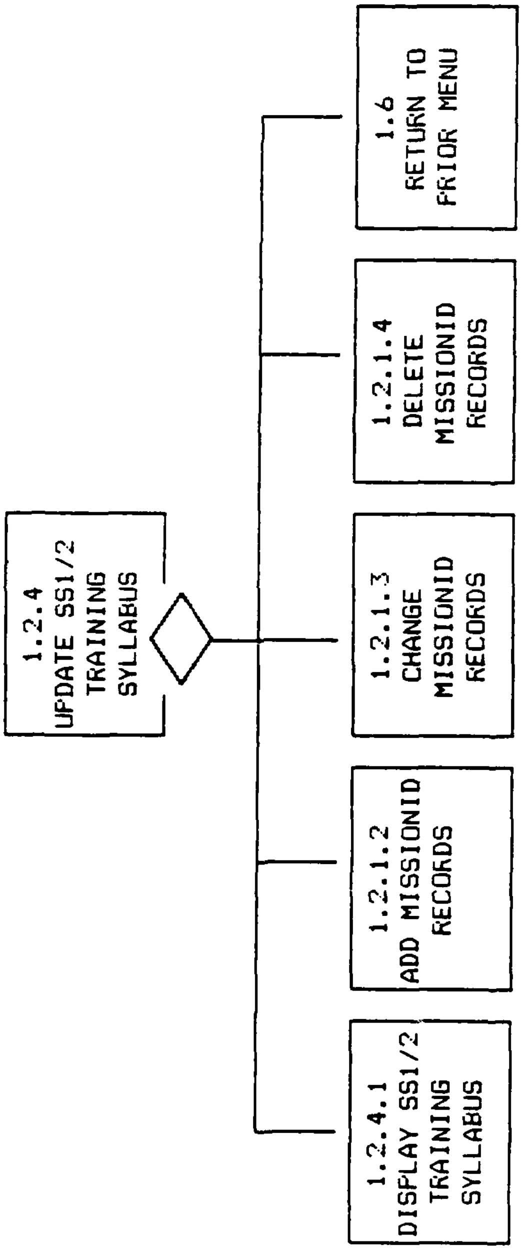
APPENDIX E: STRUCTURE CHARTS

1.2.3 UPDATE SS3 TRAINING SYLLABUS



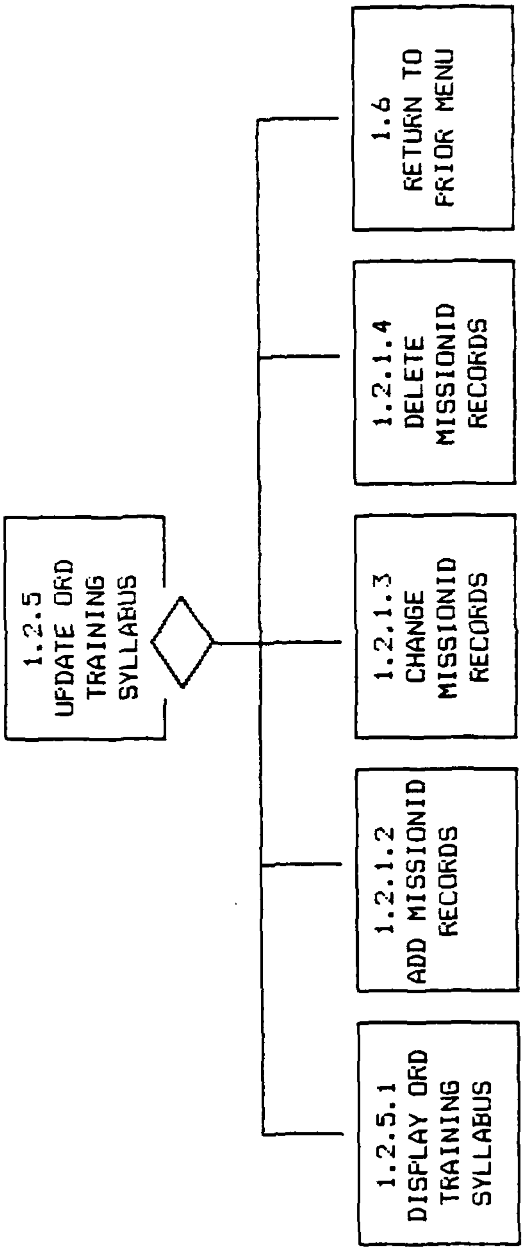
APPENDIX E: STRUCTURE CHARTS

1.2.4 UPDATE SS1/2 TRAINING SYLLABUS



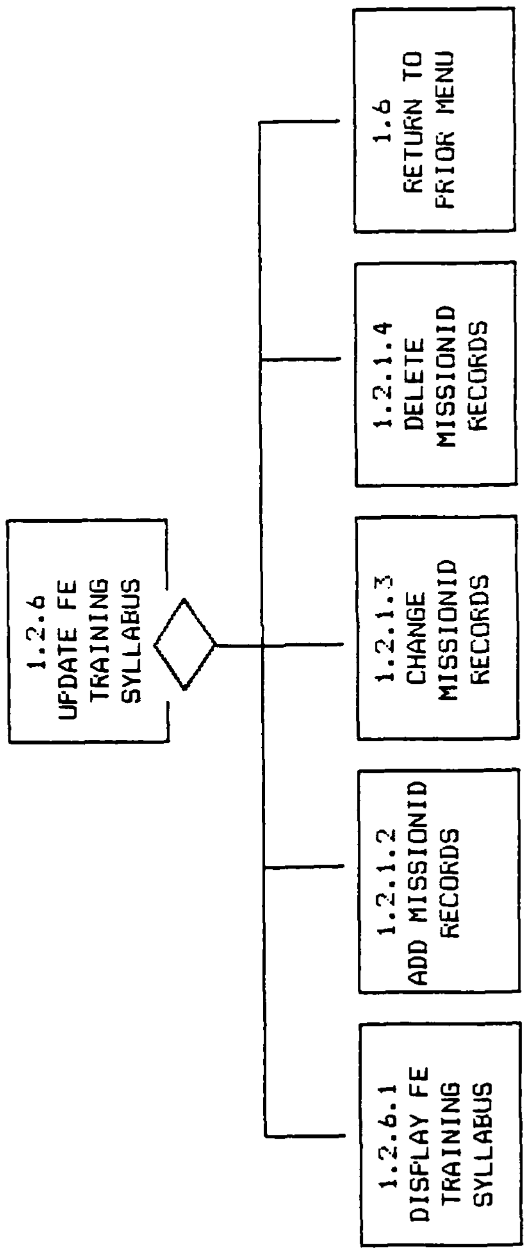
APPENDIX E: STRUCTURE CHARTS

1.2.5 UPDATE ORDNANCE TRAINING SYLLABUS


1.2.5 UPDATE ORD TRAINING SYLLABUS
1.2.5.1 DISPLAY ORD TRAINING SYLLABUS
1.2.1.2 ADD MISSIONID RECORDS
1.2.1.3 CHANGE MISSIONID RECORDS
1.2.1.4 DELETE MISSIONID RECORDS
1.6 RETURN TO PRIOR MENU

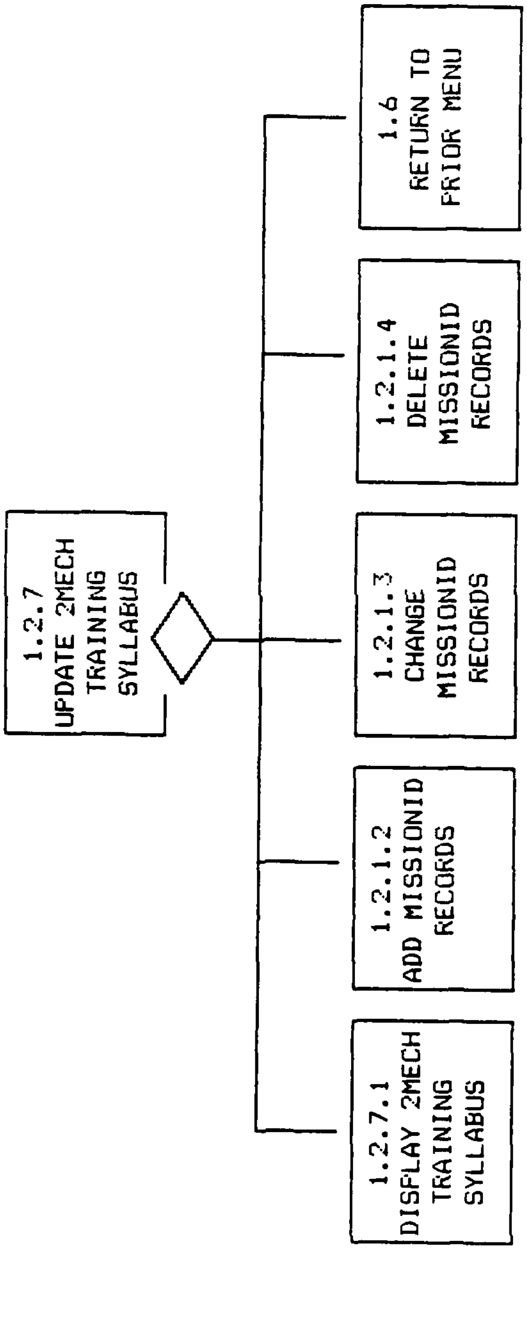
APPENDIX E: STRUCTURE CHARTS

1.2.6 UPDATE FLIGHT ENGINEER TRAINING SYLLABUS


1.2.6 UPDATE FE TRAINING SYLLABUS
1.2.6.1 DISPLAY FE TRAINING SYLLABUS
1.2.1.2 ADD MISSIONID RECORDS
1.2.1.3 CHANGE MISSIONID RECORDS
1.2.1.4 DELETE MISSIONID RECORDS
1.6 RETURN TO PRIOR MENU

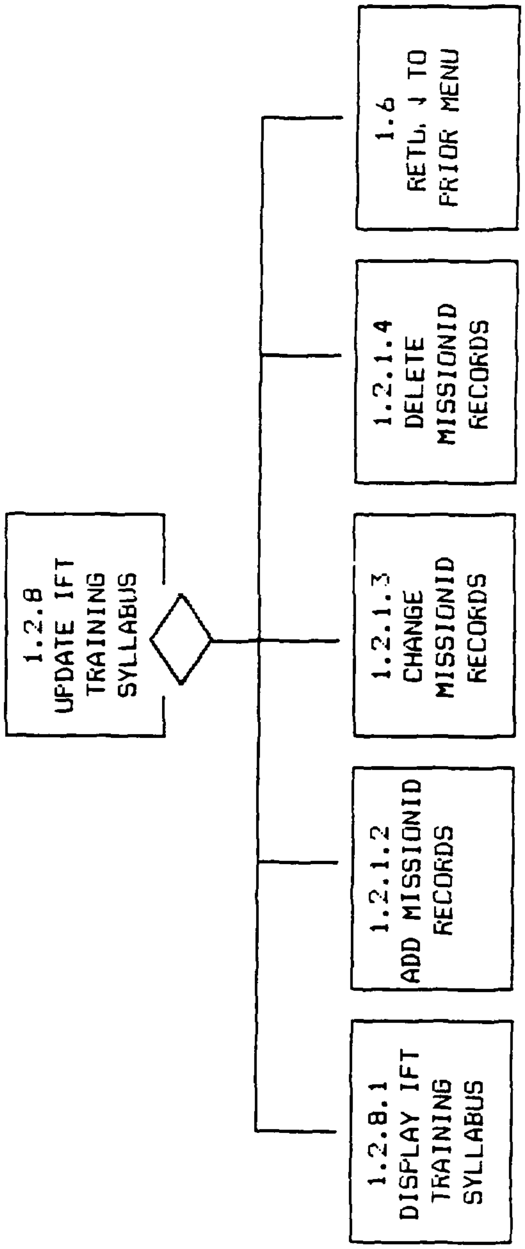
APPENDIX E: STRUCTURE CHARTS

1.2.7 UPDATE SECOND MECHANIC TRAINING SYLLABUS



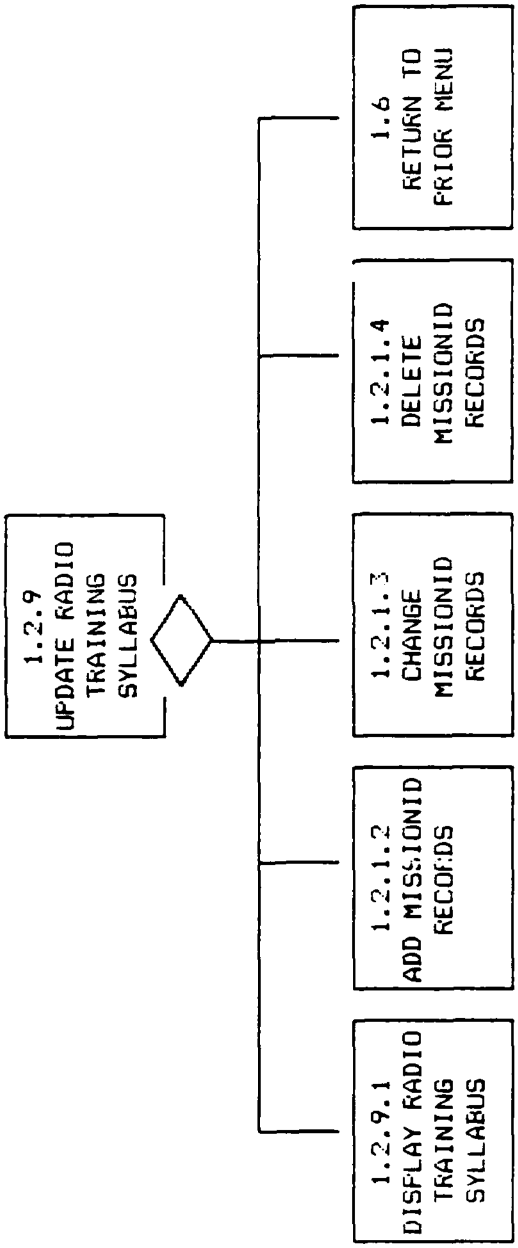
APPENDIX E: STRUCTURE CHARTS

1.2.8 UPDATE INFLIGHT TECHNICIAN TRAINING SYLLABUS



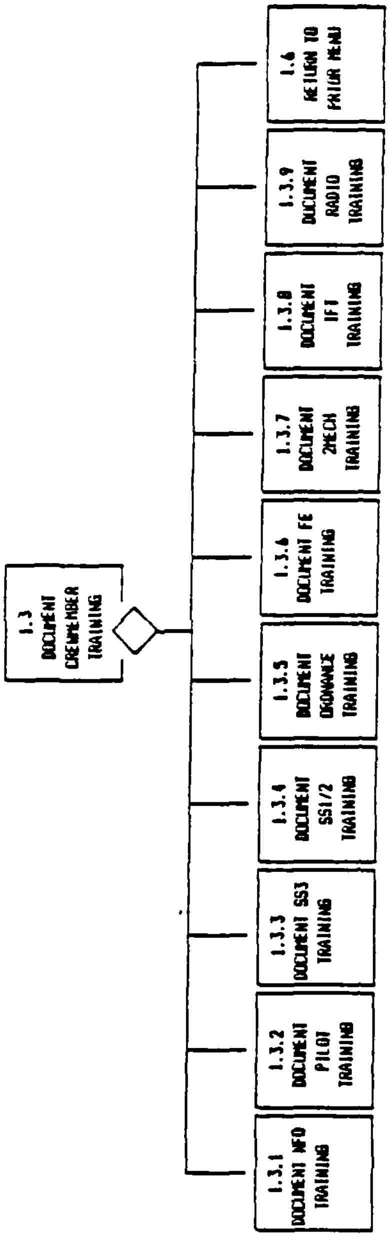
APPENDIX E: STRUCTURE CHARTS

1.2.9 UPDATE RADIO OPERATER TRAINING SYLLABUS



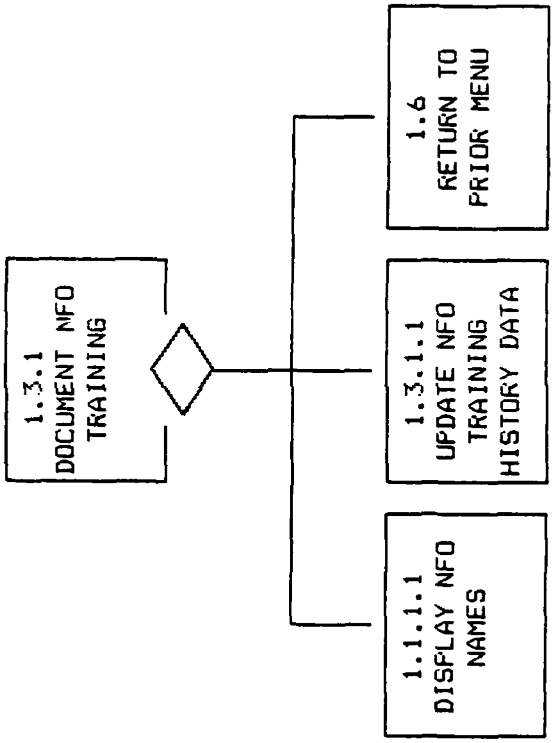
APPENDIX E: STRUCTURE CHARTS

1.3 DOCUMENT CREWMEMBER TRAINING


1.3 DOCUMENT CREWMEMBER TRAINING
1.3.1 DOCUMENT NFO TRAINING
1.3.2 DOCUMENT PILOT TRAINING
1.3.3 DOCUMENT SS3 TRAINING
1.3.4 DOCUMENT SS1/2 TRAINING
1.3.5 DOCUMENT ORDNANCE TRAINING
1.3.6 DOCUMENT FE TRAINING
1.3.7 DOCUMENT 2MECH TRAINING
1.3.8 DOCUMENT IFT TRAINING
1.3.9 DOCUMENT RADIO TRAINING
1.6 RETURN TO PRIOR MENU

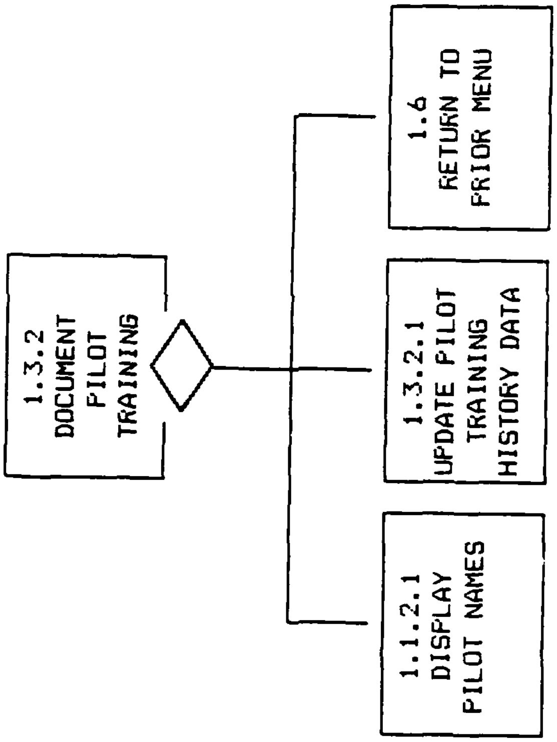
APPENDIX E: STRUCTURE CHARTS

1.3.1 DOCUMENT NFO TRAINING



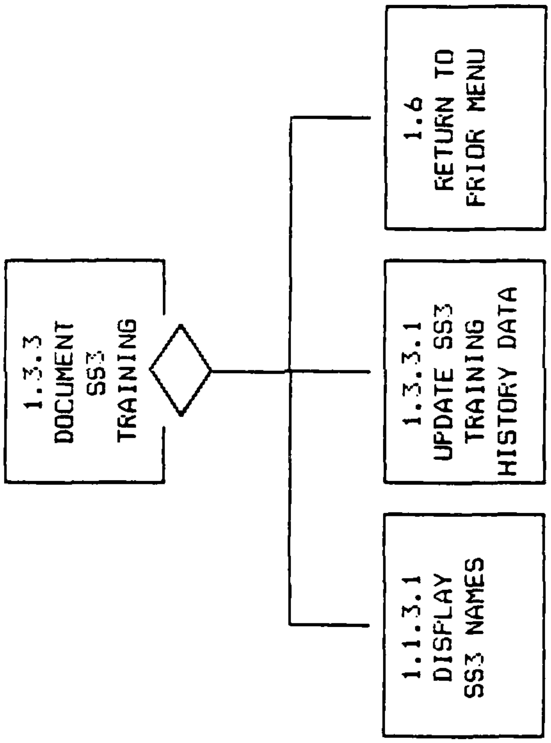
APPENDIX E: STRUCTURE CHARTS

1.3.2 DOCUMENT PILOT TRAINING



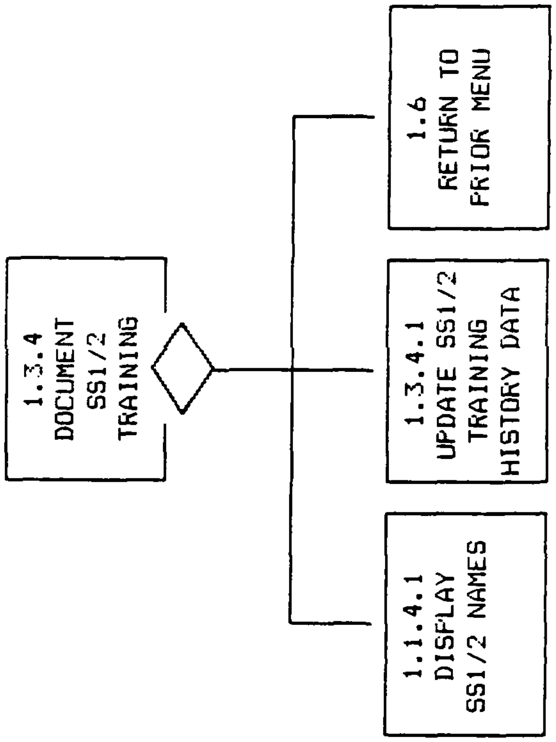
APPENDIX E: STRUCTURE CHARTS

1.3.3 DOCUMENT SS3 TRAINING


1.3.3 DOCUMENT SS3 TRAINING
1.1.3.1 DISPLAY SS3 NAMES
1.3.3.1 UPDATE SS3 TRAINING HISTORY DATA
1.6 RETURN TO PRIOR MENU

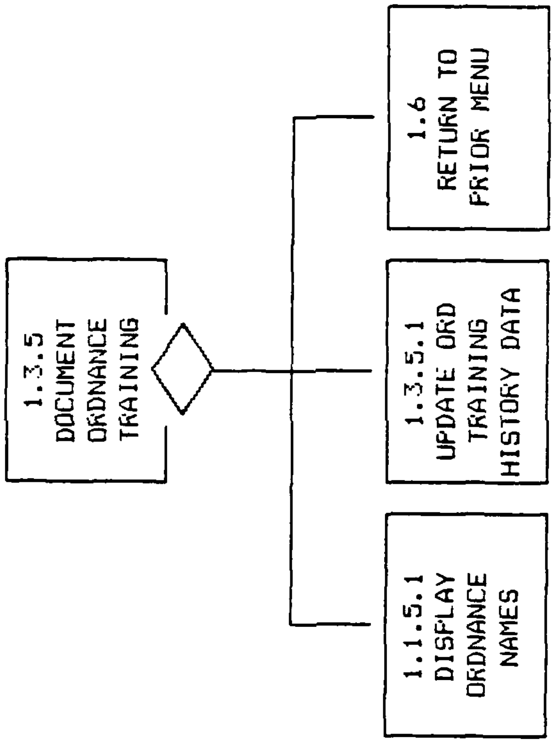
APPENDIX E: STRUCTURE CHARTS

1.3.4 DOCUMENT SS1/2 TRAINING


1.3.4 DOCUMENT SS1/2 TRAINING
1.1.4.1 DISPLAY SS1/2 NAMES
1.3.4.1 UPDATE SS1/2 TRAINING HISTORY DATA
1.6 RETURN TO PRIOR MENU

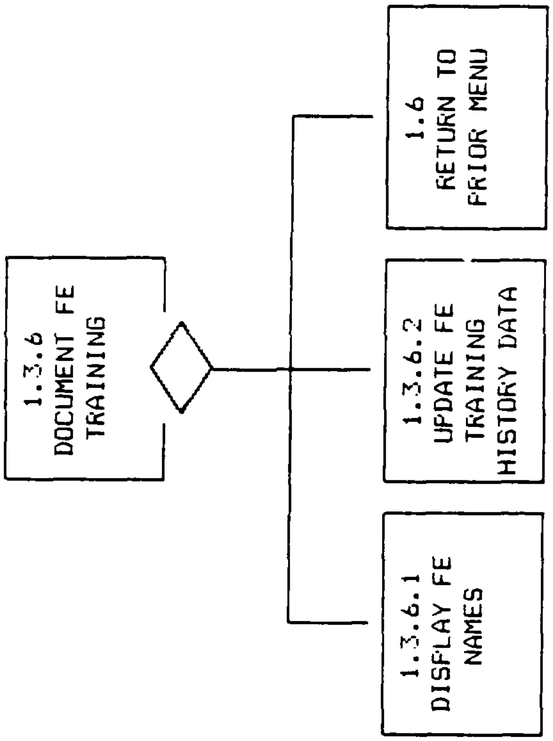
APPENDIX E: STRUCTURE CHARTS

1.3.5 DOCUMENT ORNANCEMAN TRAINING



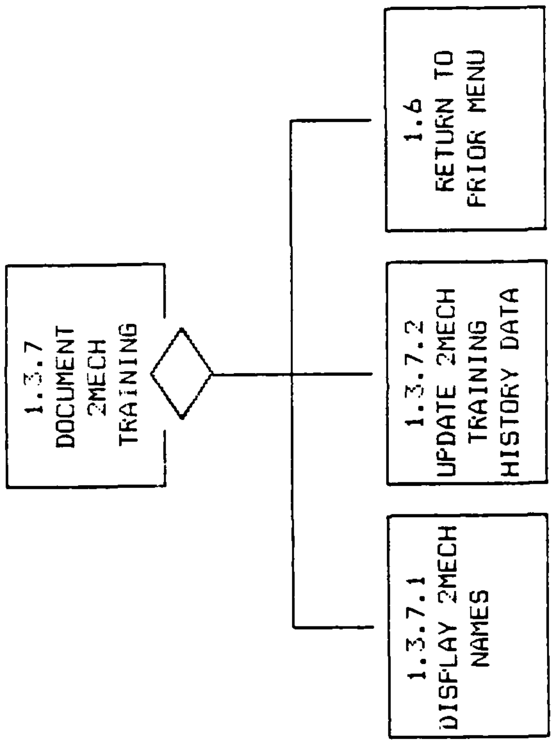
APPENDIX E: STRUCTURE CHARTS

1.3.6 DOCUMENT FLIGHT ENGINEER TRAINING


1.3.6 DOCUMENT FE TRAINING
1.3.6.1 DISPLAY FE NAMES
1.3.6.2 UPDATE FE TRAINING HISTORY DATA
1.6 RETURN TO PRIOR MENU

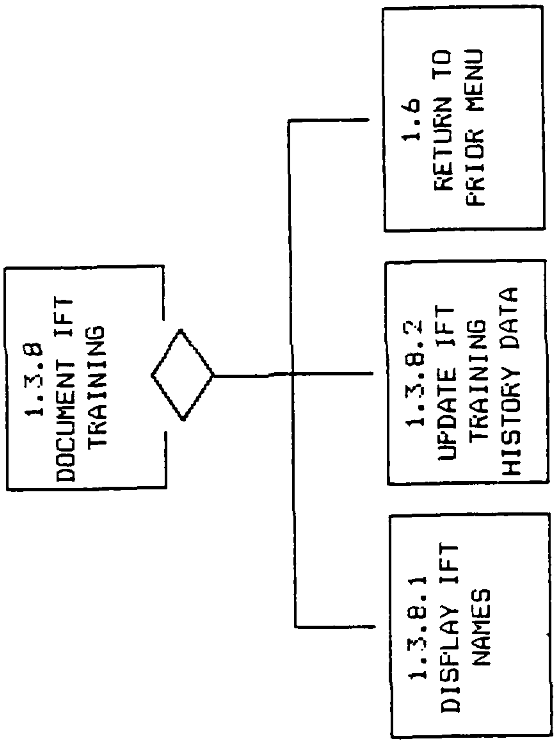
APPENDIX E: STRUCTURE CHARTS

1.3.7 DOCUMENT SECOND MECHANIC TRAINING



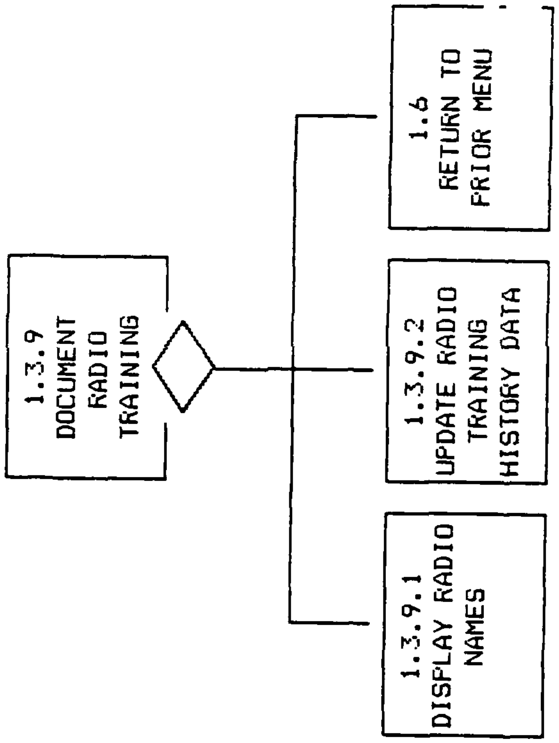
APPENDIX E: STRUCTURE CHARTS

1.3.8 DOCUMENT INFLIGHT TECHNICIAN TRAINING



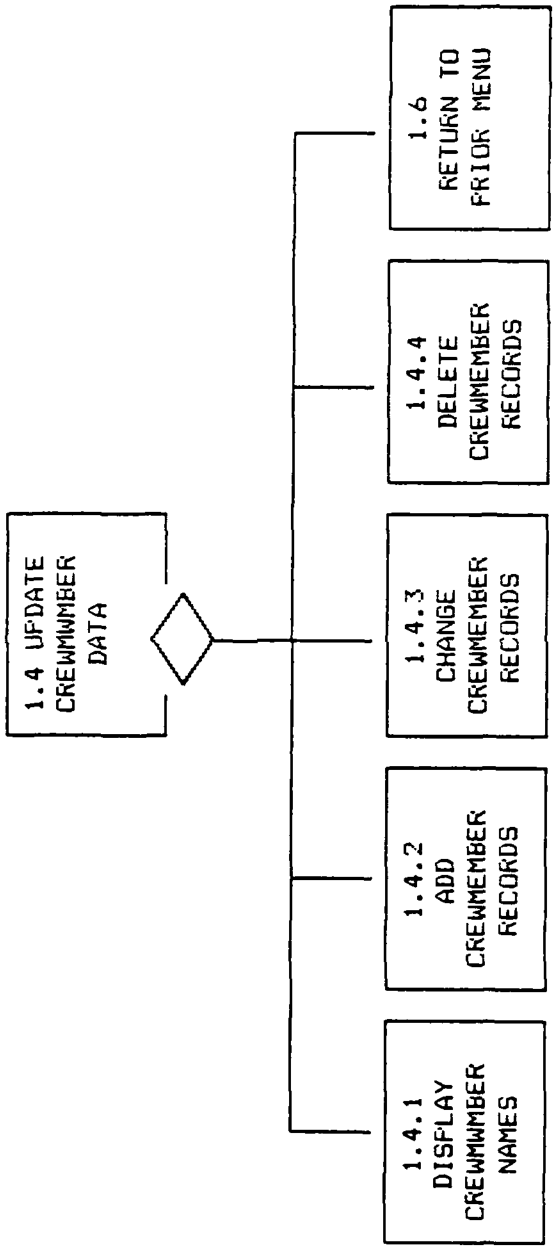
APPENDIX E: STRUCTURE CHARTS

1.3.9 DOCUMENT RADIO OPERATER TRAINING



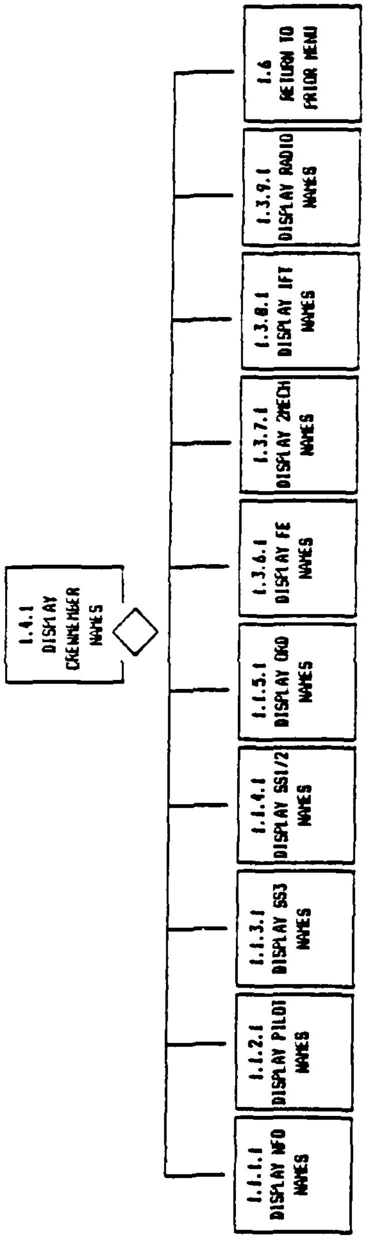
APPENDIX E: STRUCTURE CHARTS

1.4 UPDATE CREWMEMBER DATA


1.4 UFDATE CREWMWMBER DATA
1.4.1 DISFLAY CREWMWMBER NAMES
1.4.2 ADD CREWMEMBER RECORDS
1.4.3 CHANGE CREWMEMBER RECORDS
1.4.4 DELETE CREWMEMBER RECORDS
1.6 RETURN TO FRIOR MENU

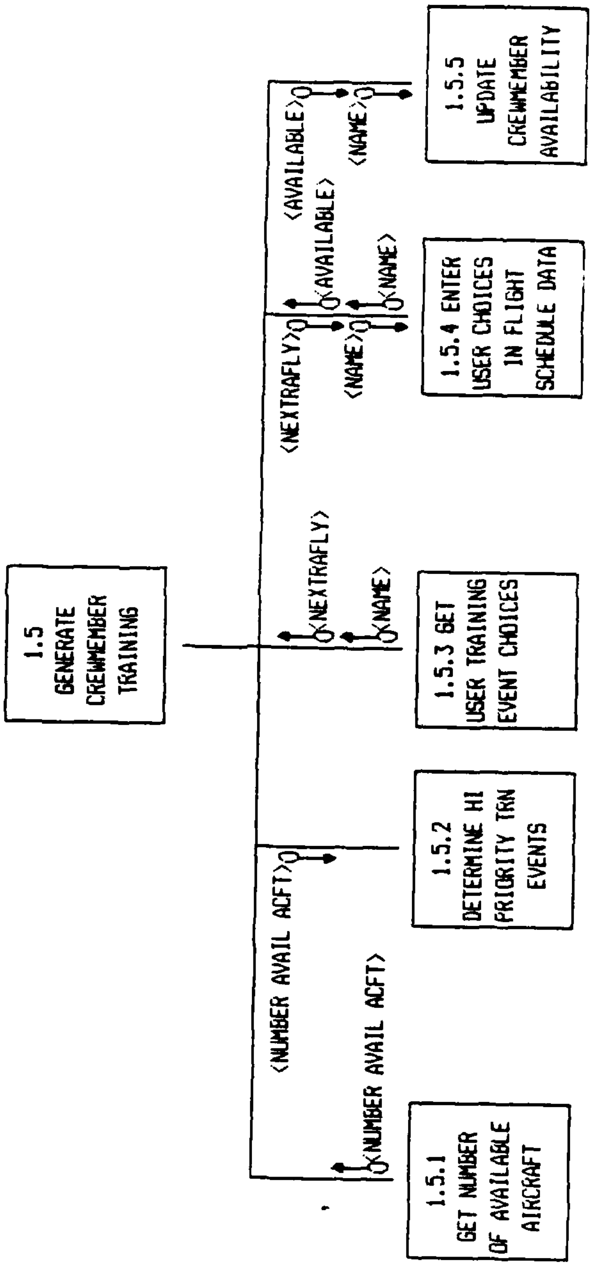
APPENDIX E: STRUCTURE CHARTS

1.4.1 DISPLAY CREWMEMBER NAMES


1.4.1 DISPLAY CREWMEMBER NAMES
1.1.1.1 DISPLAY NFO NAMES
1.1.2.1 DISPLAY PILOT NAMES
1.1.3.1 DISPLAY SS3 NAMES
1.1.4.1 DISPLAY SS1/2 NAMES
1.1.5.1 DISPLAY ORD NAMES
1.3.6.1 DISPLAY FE NAMES
1.3.7.1 DISPLAY 2MECH NAMES
1.3.8.1 DISPLAY IFT NAMES
1.3.9.1 DISPLAY RADIO NAMES
1.6 RETURN TO PRIOR MENU

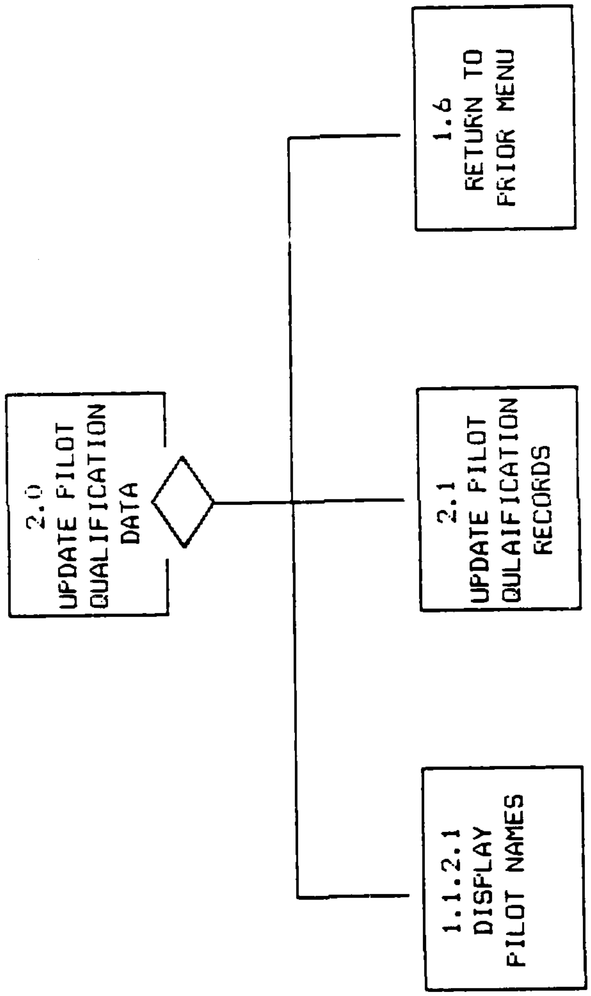
APPENDIX E: STRUCTURE CHARTS

1.5 GENERATE CREWMEMBER TRAINING


1.5 GENERATE CREWMEMBER TRAINING
1.5.1 GET NUMBER OF AVAILABLE AIRCRAFT
1.5.2 DETERMINE HI PRIORITY TRN EVENTS
1.5.3 GET USER TRAINING EVENT CHOICES
1.5.4 ENTER USER CHOICES IN FLIGHT SCHEDULE DATA
1.5.5 UPDATE CREWMEMBER AVAILABILITY
<NUMBER AVAIL ACFT>
<NUMBER AVAIL ACFT>
<NEXTRAFLY>
<NAME>
<NEXTRAFLY>
<NAME>
<AVAILABLE>
<NAME>
<AVAILABLE>
<NAME>

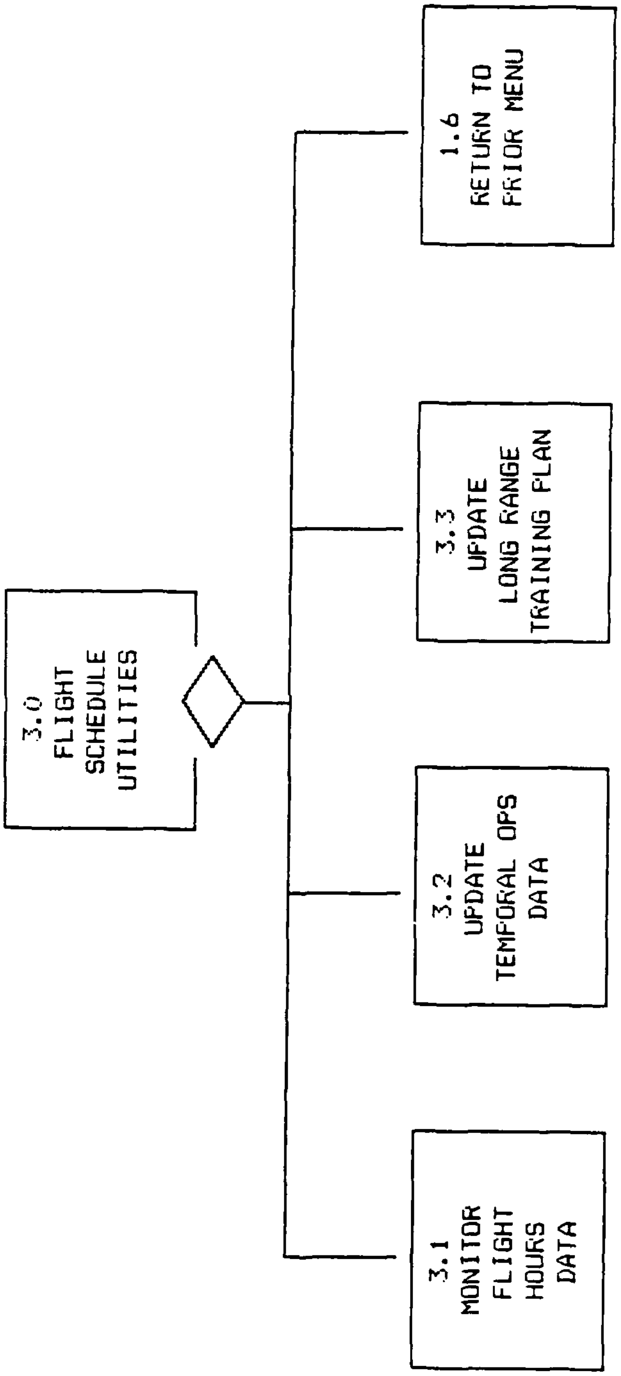
APPENDIX E: STRUCTURE CHARTS

2.0 UPDATE PILOT QUALIFICATION DATA



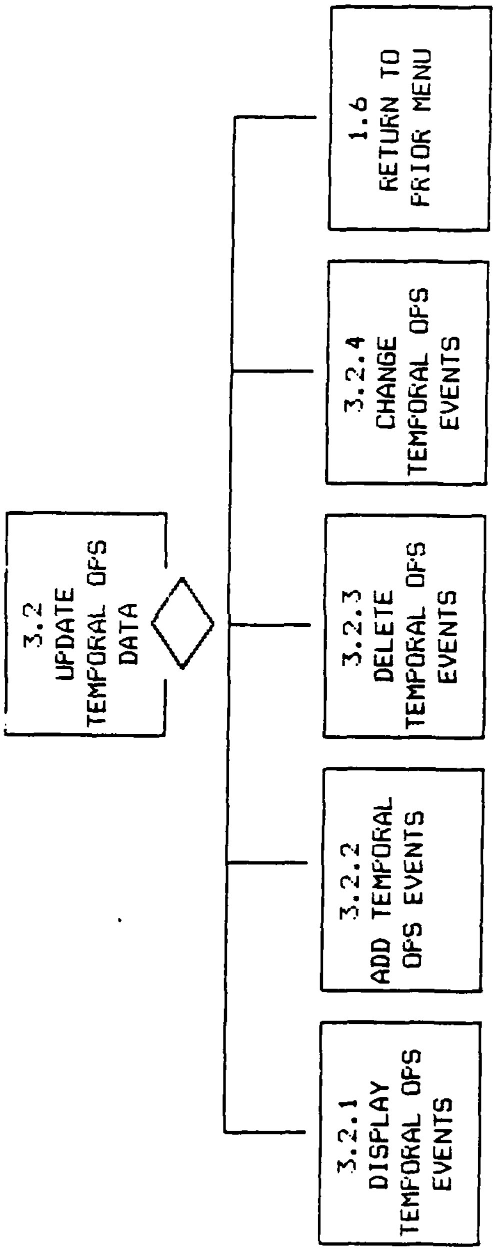
APPENDIX E: STRUCTURE CHARTS

3.0 FLIGHT SCHEDULE UTILITIES


3.0 FLIGHT SCHEDULE UTILITIES
1.6 RETURN TO PRIOR MENU
3.3 UPDATE LONG RANGE TRAINING PLAN
3.2 UPDATE TEMPORAL OPS DATA
3.1 MONITOR FLIGHT HOURS DATA

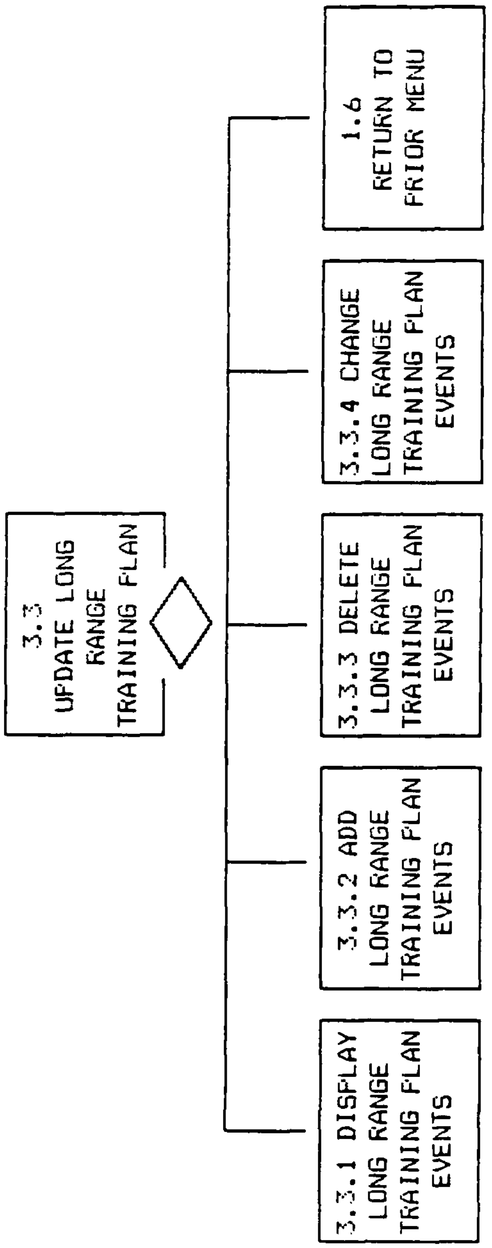
APPENDIX E: STRUCTURE CHARTS

3.2 UPDATE TEMPORAL OPERATIONS DATA


3.2 UPDATE TEMPORAL OPS DATA
3.2.1 DISPLAY TEMPORAL OPS EVENTS
3.2.2 ADD TEMPORAL OPS EVENTS
3.2.3 DELETE TEMPORAL OPS EVENTS
3.2.4 CHANGE TEMPORAL OPS EVENTS
1.6 RETURN TO PRIOR MENU

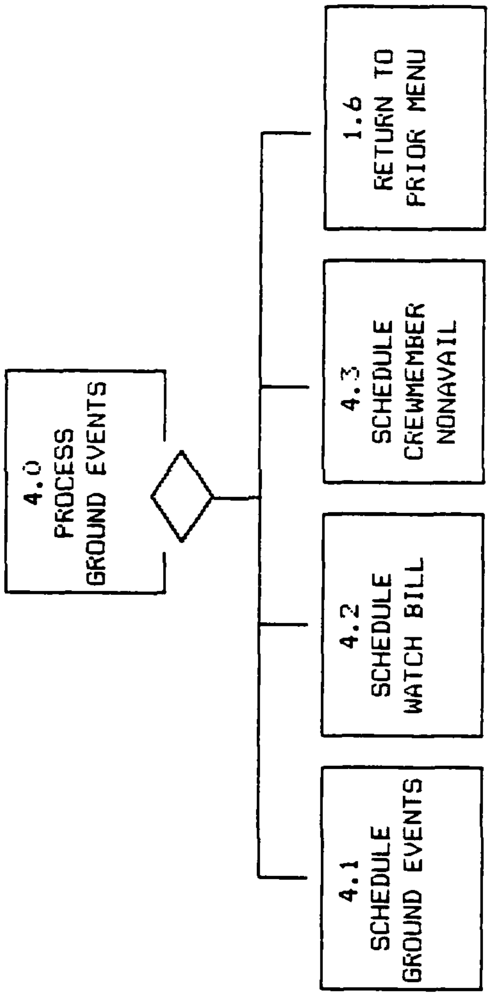
APPENDIX E: STRUCTURE CHARTS

3.3 UPDATE LONG RANGE TRAINING PLAN


3.3 UPDATE LONG RANGE TRAINING PLAN
3.3.1 DISPLAY LONG RANGE TRAINING PLAN EVENTS
3.3.2 ADD LONG RANGE TRAINING PLAN EVENTS
3.3.3 DELETE LONG RANGE TRAINING PLAN EVENTS
3.3.4 CHANGE LONG RANGE TRAINING PLAN EVENTS
1.6 RETURN TO PRIOR MENU

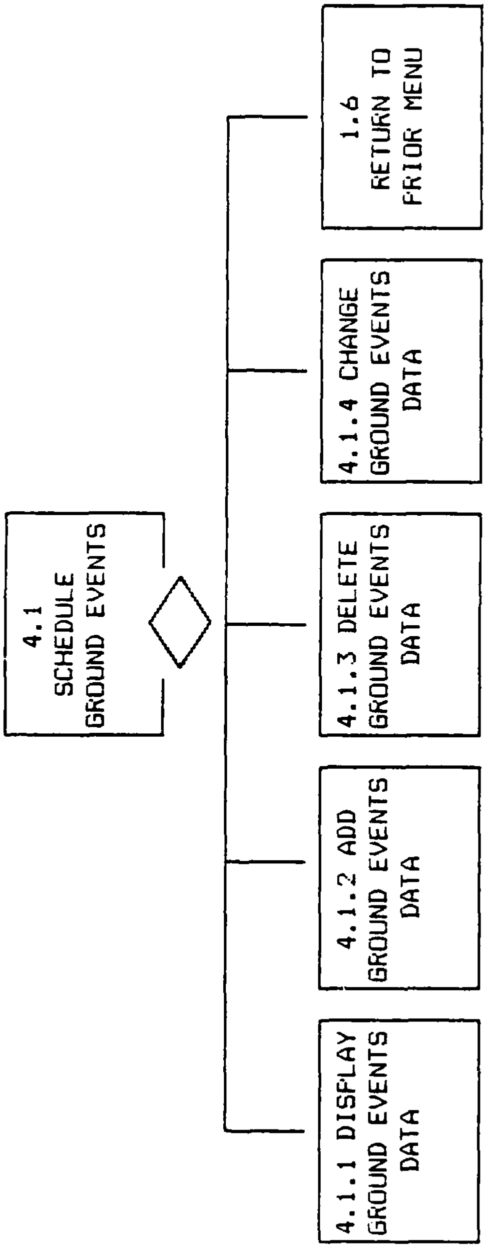
APPENDIX E: STRUCTURE CHARTS

4.0 PROCESS GROUND EVENTS


4.0 PROCESS GROUND EVENTS
1.6 RETURN TO PRIOR MENU
4.3 SCHEDULE CREWMEMBER NONAVAIL
4.2 SCHEDULE WATCH BILL
4.1 SCHEDULE GROUND EVENTS

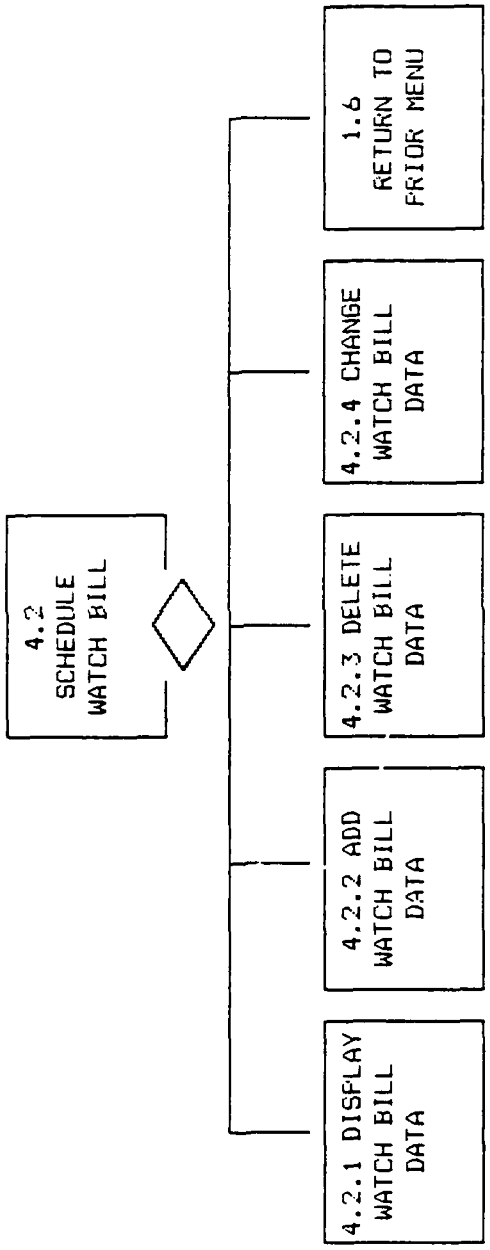
APPENDIX E: STRUCTURE CHARTS

4.1 SCHEDULE GROUND EVENTS


4.1 SCHEDULE GROUND EVENTS
4.1.1 DISPLAY GROUND EVENTS DATA
4.1.2 ADD GROUND EVENTS DATA
4.1.3 DELETE GROUND EVENTS DATA
4.1.4 CHANGE GROUND EVENTS DATA
1.6 RETURN TO PRIOR MENU

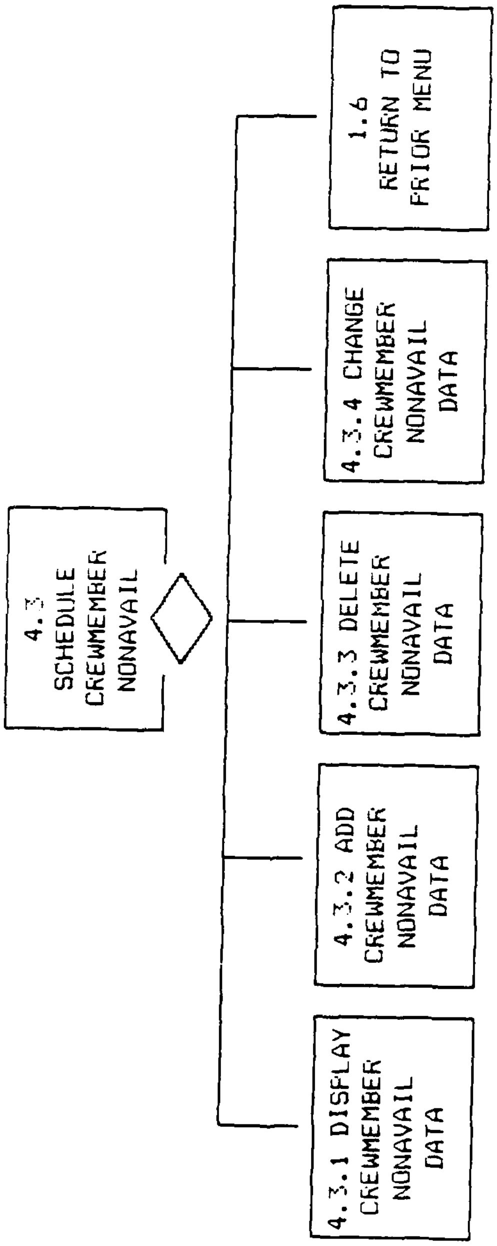
APPENDIX E: STRUCTURE CHARTS

4.2 SCHEDULE WATCH BILL


4.2 SCHEDULE WATCH BILL
4.2.1 DISPLAY WATCH BILL DATA
4.2.2 ADD WATCH BILL DATA
4.2.3 DELETE WATCH BILL DATA
4.2.4 CHANGE WATCH BILL DATA
1.6 RETURN TO PRIOR MENU

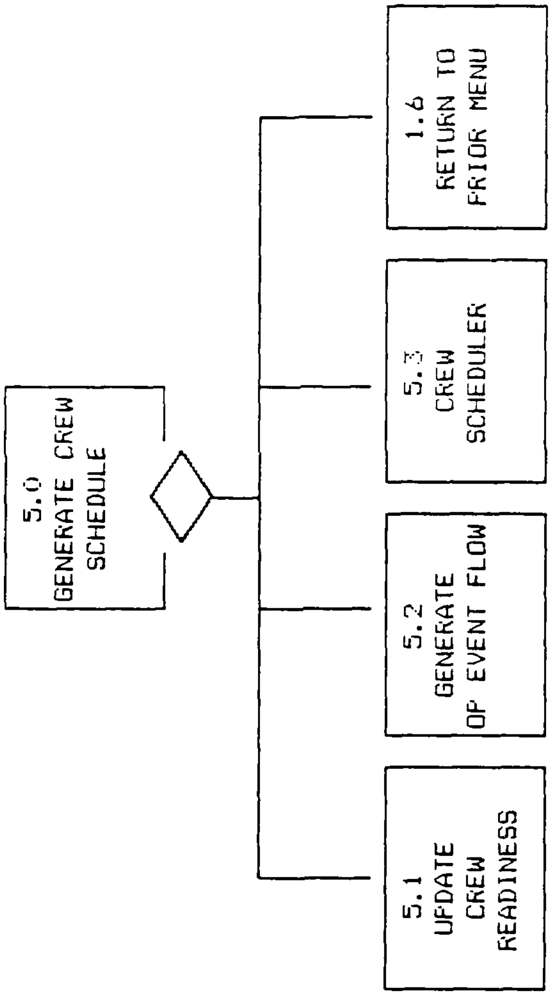
APPENDIX E: STRUCTURE CHARTS

4.3 SCHEDULE CREWMEMBER NON-AVAILABILITY


4.3 SCHEDULE CREWMEMBER NONAVAIL
4.3.1 DISPLAY CREWMEMBER NONAVAIL DATA
4.3.2 ADD CREWMEMBER NONAVAIL DATA
4.3.3 DELETE CREWMEMBER NONAVAIL DATA
4.3.4 CHANGE CREWMEMBER NONAVAIL DATA
1.6 RETURN TO PRIOR MENU

APPENDIX E: STRUCTURE CHARTS

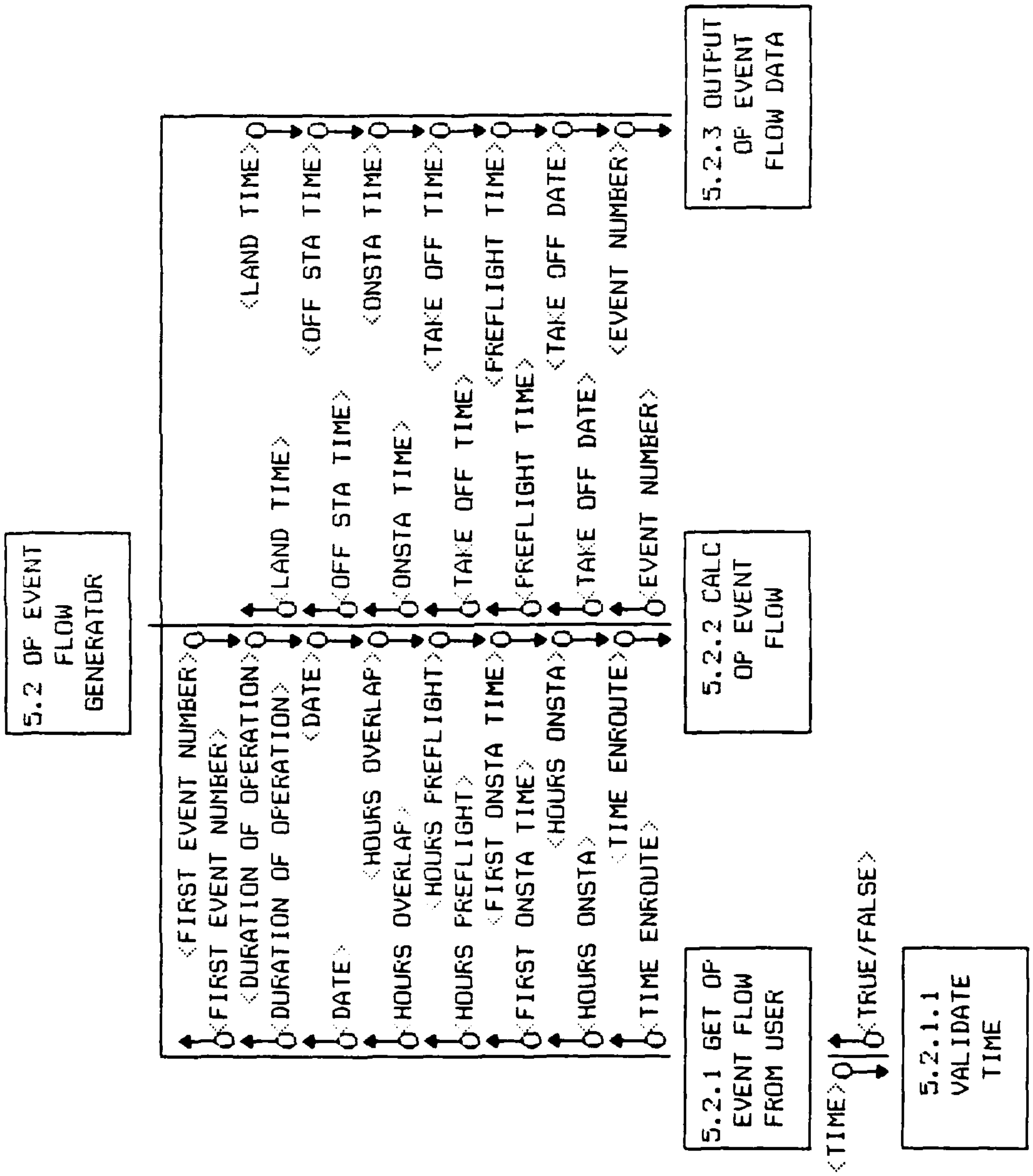
5.0 GENERATE CREW SCHEDULE



5.0 GENERATE CREW SCHEDULE
1.6 RETURN TO PRIOR MENU
5.3 CREW SCHEDULER
5.2 GENERATE OP EVENT FLOW
5.1 UPDATE CREW READINESS

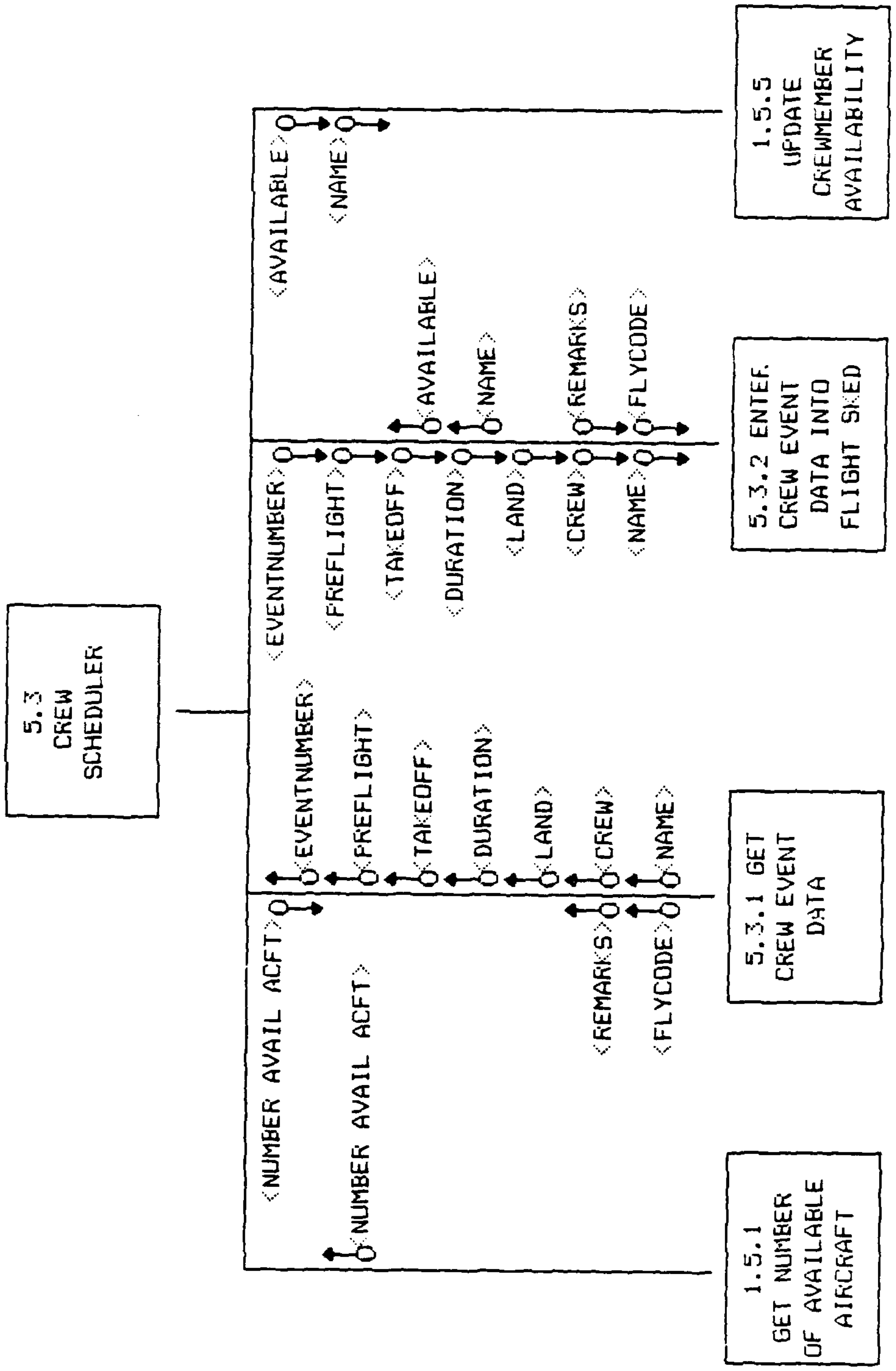
APPENDIX E: STRUCTURE CHARTS

5.2 OPERATIONAL EVENT FLOW GENERATOR



APPENDIX E: STRUCTURE CHARTS

5.3 CREW SCHEDULER



APPENDIX F: SAMPLE REPORT FORMATS

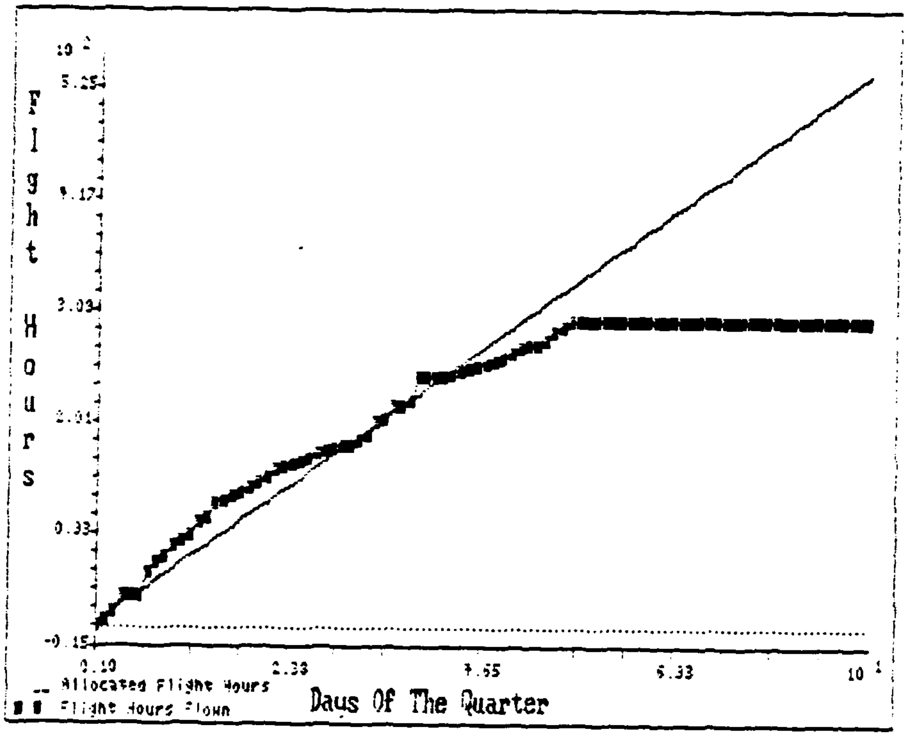
EXAMPLE FLIGHT SCHEDULE

```
-------------------------------------------------------------
Commanding Officer's Name             Executive Officers Name
*************************************************************
Date                                              Julian Date
SDO: RANK NAME                          ASDO: 08-16 RATE NAME
Taxi Pilot: RANK NAME                         16-24 RATE NAME
Duty FE: RATE NAME                            24-08 RATE NAME
Duty Crew: CREWNUMBER                   Sonolocker: DATE NAME
CMS: BEGIN-END RATE NAME
     BEGIN-END RATE NAME
-------------------------------------------------------------
Event        **Mission Commander    Crew Number   Type Flight
Brf/Pft      *PPC/TC                Deletions     Flt Code
Takeoff      Additions                            Notes
Duration
Land
-------------------------------------------------------------
EVENTNUMBER  **RANK, NAME           CREWNUMBER    TYPEFLT
PREFLIGHT    *RANK, NAME            DELNAME1      FLTCODE
TAKEOFF      ADDNAME1               DELNAME2      NOTENUM1
DURATION     ADDNAME2               DELNAME3      NOTENUM2
LAND         ADDNAME3               DELNAME4
             ADDNAME4               DELNAME5
             ADDNAME5
-------------------------------------------------------------
*************************************************************
                       GROUND TRAINING
TIME         NAME                  EVENT          PLACE    NOTES
-------------------------------------------------------------
BEGIN-END    RANK,NAME             EVENT          PLACE    NOTE-
                                                           NUMS
-------------------------------------------------------------
NOTE 1
NOTE 2
NOTE 3
 . . .
-------------------------------------------------------------
*************************************************************
```

APPENDIX F: SAMPLE REPORT FORMATS

EXAMPLE FLIGHT HOURS MONITORING GRAPH

Flight Hour Progress Graph



LIST OF REFERENCES

1. Davis, William S., Systems Analysis and Design, Addison-Wesley, 1983

2. Page-Jones, Meilir, The Practical Guide to Structured Systems Design, Yourdon Press, 1980.

3. Davis, Gordon B., and Olson, Margrethe H., Management Information Systems: Conceptual Foundations, Structure, and Development, McGraw-Hill, 1985.

4. Kroenke, David M., Database Processing: Fundamentals, Design, Implementation, Science Research, 1983.

5. Cesena, Mary L., and Jones, Wendell O. Col., USA., "Accelerated Information Systems Development in the Army", Society for Information Management Spectrum, September 1987.

6. Pressman, Roger S., Software Engineering: A Practioner's Approach, McGraw-Hill, 1987.

LIST OF REFERENCES

7. Bennett, John L., Building Decision Support Systems, Addison-Westley, 1983.

8. Rowe, Neil C., "Class Notes for CS3310, Artificial Intelligence, 1987", Naval Postgraduate School, Monterey, California.

9. Townsend, Carl, Introduction to Turbo Prolog, Sybex, 1986.

10. Yourdon, Edward, Managing the Structured Techniques: Strategies for Software Development in the 1990's, Yourdon Press, 1986.

11. Boehm, Barry W., Software Engineering Economics, Prentice-Hall, 1981.

12. DeMarco, Tom, Controlling Software Projects, Yourdon Press, 1982.

BIBLIOGRAPHY

Abdel-Hamid, Tarek K., and Madnick, Stuart E., "Managing Software Projects: An Integrative System Dynamics Modeling Approach", Naval Postgraduate School, Monterey, California, Working Paper No. 87-14, June 1987.

Bennett, John L., Builaing Decision Support Systems, Menlo Park: Addison-Westley, 1983.

Boehm, Barry W., Software Engineering Economics, Englewood Cliffs: Prentice-Hall, 1981.

Carrabis, Joseph David, dBase III Advanced Programming, Indianapolis: Que, 1985.

Carrabis, Joseph David, dBase III Plus: The Complete Reference, Berkeley: Osborne McGraw-Hill, 1987

Cesena, Mary L., and Jones, Wendell O., Col., USA., "Accelerated Information Systems Development in the Army", Society for Information Management Spectrum, September, 1987.

BIBLIOGRAPHY

Davis, Gordon B., and Olson, Margrethe H., Management Information Systems: Conceptual Foundations, Structure, and Development, San Francisco: McGraw-Hill, 1985.

Davis, William S., Systems Analysis and Design, Menlo Park: Addison-Wesley, 1983

DeMarco, Tom, Controlling Software Projects, Englewood Cliffs: Yourdon Press, 1982.

Koffman, Elliot B., Problem Solving and Structured Programming in Pascal, Menlo Park: Addison-Wesley, 1985.

Kroenke, David M., Database Processing: Fundamentals, Design, Implementation, Chicago: Science Research, 1983.

McMenamin, Stephen M. and Palmer, John F, Essential Systems Analysis, New York: Yourdon Press, 1984.

Page-Jones, Meilir, The Practical Guide to Structured Systems Design, Englewood Cliffs: Yourdon Press, 1980.

BIBLIOGRAPHY

Pressman, Roger S., Software Engineering: A Practioner's Approach, San Francisco: McGraw-Hill, 1987.

Rowe, Neil C., "Class Notes for CS3310, Artificial Intelligence, 1987", Naval Postgraduate School, Monterey, California.

Simpson, Allen, Simpson's dBase Tips and Tricks, San Francisco: Sybex, 1987.

Townsend, Carl, Introduction to Turbo Prolog, San Francisco: Sybex, 1986.

Townsend, Carl, Mastering dBase III Plus: A Structured Approach, San Francisco: Sybex, 1986.

Winston, Patrick Henry, Artificial Intelligence, Menlo Park: Addison-Wesley, 1984.

Yourdon, Edward, Managing the Structured Techniques: Strategies for Software Development in the 1990's, New York: Yourdon Press, 1986.

INITIAL DISTRIBUTION LIST

No. Copies

1. Chief of Naval Operations 2
   Director, Information Systems (OP-945)
   Navy Department
   Washington, D.C. 20350-2000

2. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

3. Library, Code 0142 2
   Naval Postgraduate School
   Monterey, California 93943-5002

4. Barry A. Frew, Code 54Fw 1
   Department of Adminstrative Sciences
   Naval Postgraduate School
   Monterey, California 93943-5002

5. John Isett, Code 54Is 1
   Department of Administrative Sciences
   Naval Postgraduate School
   Monterey, California 93943-5002

INITIAL DISTRIBUTION LIST

6. Commander, Naval Air Forces U.S. Pacific Fleet 3
Sam Anderson, Code 23
Naval Air Station North Island
San Diego, California 92135

7. Commander, Patrol Wings Pacific 3
Operations Officer
Naval Air Station
Moffett Field, California 94035

8. Commander, Patrol Wings Atlantic 3
Operations Officer
Naval Air Station
Brunswick, Maine 04001

9. Commander, Naval Air Systems Command 10
Department of the Navy (APC-209)
Washington, D.C. 20361-1209

END
DATE
FILMED
8-88
DTIC